Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Anže Mikec

Razlikovanje normalnih in rakavih urotelijskih celic iz mikroskopskih slik z uporabo strojnega učenja

magistrsko delo

magistrski program druge stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Janez Demšar
Somentor: izr. prof. dr. Mateja Erdani Kreft

Ljubljana, 2016

Izjava o avtorstvu magistrskega dela

Spodaj podpisani Anže Mikec sem avtor magistrskega dela z naslovom:

Razlikovanje normalnih in rakavih urotelijskih celic iz mikroskopskih slik z uporabo strojnega učenja

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Janez Demšar in somentorstvom izr. prof. dr. Mateja Erdani Kreft,
- so elektronska oblika magistrskega dela, naslov (slovenski, angleški), povzetek (slovenski, angleški) ter ključne besede (slovenske, angleške) identični s tiskano obliko magistrskega dela,
- soglašam z javno objavo elektronske oblike magistrskega dela v zbirki "Dela FRI".

V Ljubljani, 28. septembra 2016 Podpis avtorja:

Zahvala

Iskrena hvala mentorju izr. prof. dr. Janezu Demšarju za vso pomoč, znanje in izvrstne ideje. Zahvaljujem se somentorici izr. prof. dr. Mateji Erdani Kreft za usmeritev, natančnost in zagnanost. Hvala doc. dr. Veroniki Kloboves Prevodnik za čas, pomoč in citopatološke urinske vzorce.

Prisrčna hvala staršema za podporo in potrpežljivost, kolegu in prijatelju Mitji za vztrajnost in vzpodbudo, Neži za modre misli ter vsem ostalim, ki ste mi stali ob strani.

Anže Mikec, 2016

Kazalo

Povzetek

Mnogi raziskovalci in zdravniki zaradi pomanjkanja dobrih in zanesljivih orodij, mikroskopske slike s celicami še vedno označujejo ročno, kar je časovno potratno in zvišuje stroške raziskovanja in zdravljenja. Kot odgovor na ta problem, smo razvili orodje, ki z metodo Watershed samodejno zaznava in razločuje normalne in rakave urotelijske celice. Orodje v prvih korakih segmentira mikroskopske slike in označi regije odkritih celic. Na osnovi odkritih celic sledi izbor in izračun značilk, ki jih orodje v naslednjih korakih uporabi za učenje napovednih modelov. V raziskavi smo celice v ciljna razreda uvrščali z nevronskimi mrežami, naključnimi gozdovi, naivnim Bayesovim klasifikatorjem, odločitvenimi pravili CN2, metodo podpornih vektorjev ter metodama boosting in bagging. Opisan postopek smo izvedli s samodejno označenimi, nato pa še z ročno označenimi slikami normalnih prašičjih in rakavih humanih urotelijskih celic. Empirična opazovanja kažejo, da orodje dobro segmentira celice. Kljub temu se izkaže, da napovedni modeli boljše razločujejo med normalnimi in rakavimi celicami na ročno označenih celicah. Najboljše rezultate z ročno označenimi celicami dosegajo nevronske mreže (AUC (*area under the curve*) 0,9052), metoda bagging (AUC 0,9041) in naključni gozdovi (AUC 0,9005). Zmogljivost orodja smo preverili še z naborom citopatoloških urinskih vzorcev. Pri teh vzorcih so rezultati samodejne segmentacije opazno slabši kot pri drugih naborih slik. Kljub temu, bi lahko z nadaljnjimi izboljšavami orodje bistveno pripomoglo k poenostavljenim in zanesljivejšim analizam mikroskopskih slik rakavih celic.

Ključne besede

rakave celice, rak na sečnem mehurju, razločevanje normalnih in rakavih urotelijskih celic, segmentacija slik, obdelava mikroskopskih slik, segmentacija mikroskopskih slik, strojno učenje, morfologija celic

Abstract

As a result of a lack of reliable tooling, much of the cell detection in microscopic imaging is still done manually. This in turn raises research and treatment costs. To tackle this problem, we developed a tool, which automatically detects and classifies normal and cancerous urothelial cells. In the first part the tool segments microscopic images and marks the discovered cell regions. On the basis of the discovered regions, the tool extracts a set of features, which are later used for learning classification models. Neural nets, random forests, naive Bayes classificator, decision rules, SVM, boosting and bagging were used for classification. We used both automatically and manually marked images of normal pig cells and cancerous human cells. Empirical observation shows, that the tool segments cells really well, nonetheless, we noticed that classificators perform better on manually marked cells. The best results were achieved (using manually marked cells) by neural nets (AUC (*area under the curve*) 0,9052), bagging (AUC 0,9041) and random forests (AUC 0,9005). The performance of the tool was further tested with cytopathological urine samples. The results of image segmentation with these samples were noticeably worse than with other image sets. With future enhancements this tool could considerably contribute to simpler and more reliable microscopic image analysis of cancerous cells.

Keywords

cancerous cells, urothelial cancer, cancer cell classification, image segmentation, microscope image processing, microscope image segmentation, machine

learning, cell morphology

Slike

Tabele

Tabela kratic in izrazov

AUC	Površina pod krivuljo (Area under the curve).
CA	Napovedna točnost (classification accuracy).
CP1	Množica celic označenih s programom CellProfiler.
CP2	Množica celic označenih s programom CellProfiler, brez celic, ki mejijo na rob slike.
ICO	Celični indeks (indeks celične oblike).
in vivo	Besedna zveza, ki se nanaša na procese, ki potekajo v živem organizmu.
in vitro	Besedna zveza, ki se nanaša na procese, ki potekajo v nadzorovanem okolju zunaj živega organizma.
MD	Mahalanobisova razdalja (Mahalanobis distance)
NMIBC	Mišično neinvazivna oblika raka sečnega mehurja (non-muscle-invasive bladder cancer).
PCA	Metoda glavnih komponent (principal component analysis).
plazmalema	Celična membrana.
proliferacija	Rast populacije celic z delitvijo.
R	Množica ročno označenih celic.
ROC	Graf, ki prikazuje razmerje deleža resnično pozitivnih in lažno pozitivnih primerov (Receiver operating characteristic).
SVM	Metoda podpornih vektorjev (Support vector machines).
U	Množica ročno označenih celic s slik urinskih vzorcev.

Poglavje 1

Uvod

1.1 Motivacija

Mišično neinvazivna oblika raka sečnega mehurja je najpogostejša oblika raka sečnega mehurja. Zaradi pogostosti in visoke verjetnosti ponovitve so stroški zdravljenja visoki [1]. Veliko vlogo pri nastanku in napredovanju raka sečnega mehurja v mišično invazivno obliko igra izguba sposobnosti celične diferenciacije urotelija. Status diferenciacije karcinoma urotelija se lahko oceni s histopatološkim pregledom. Ocena tipično sledi morfološkemu vrednotenju. Prisotnost in stopnjo tumorja lahko ocenimo tudi z uporabo bioloških označevalcev (biomarkerjev) [2]. Biomarkerji, ki služijo razločevanju normalnih in rakavih celic, so znani in zadovoljivo zanesljivi. Analizo mikroskopskih slik celic kljub tehnološkemu napredku patologi in raziskovalci pogosto opravljajo ročno, kar je časovno potratno in zvišuje stroške raziskovanja in zdravljenja.

Cilj magistrske naloge je razviti algoritem, ki bo samodejno, učinkovito in zanesljivo razločeval normalne in rakave urotelijske celice. Algoritem bo pripomogel k avtomatizaciji procesa označevanja, izboljšanju stopnje razpoznave, razbremenitvi raziskovalcev in zdravnikov, boljšemu razumevanju povezav med kancerogenostjo in morfološkimi lastnostmi celic ter znižanju

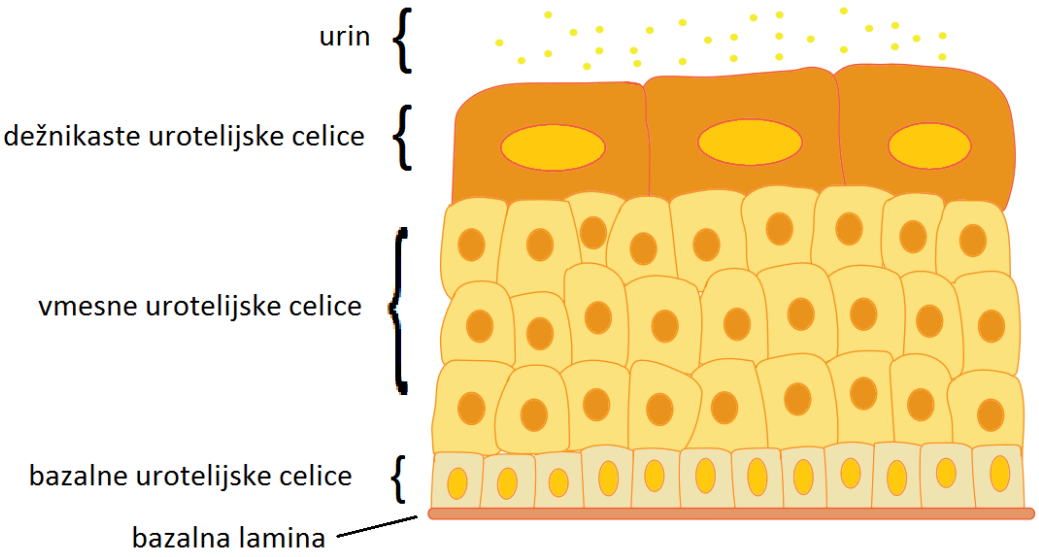
stroškov raziskovanja in zdravljenja. Čeprav se naloga osredotoča na delo z rakom na sečnem mehurju, je pomemben prispevek uporaba in prilagoditev obstoječih metod računalniškega vida, strojnega učenja pri analizi celic, ki se lahko aplicira tudi na ostalih področjih raziskovanja rakavih obolenj.

1.2 Struktura naloge

V prvem poglavju glavnega sklopa magistrskega dela podamo problematiko raka sečnega mehurja in opišemo njegove posebnosti. Na kratko se dotaknemo vidikov, ki motivirajo raziskovanje na tem področju in tudi tistih, ki to otežujejo. Naslednje poglavje služi pregledu metod in orodij, ki smo jih uporabili v tej magistrski nalogi. Opišemo mere in značilke, ki so po našem mnenju ključne pri razločevanju normalnih in rakavih celic. Vključene so tako metode računalniške analize slik in strojnega učenja kot tudi programsko opremo, ki smo jo uporabili. V tretjem sklopu podamo podrobnejši opis postopka za razločevanje normalnih in rakavih celic, njegove korake ter uporabo konceptov in orodij, ki smo jih opisali v prejšnjem sklopu. Nadaljujemo s kritičnim vrednotenjem rezultatov, kjer podamo pričakovanja in argumentacijo rezultatov, ki smo jih pridobili z našo metodo. V zadnjem, končnem poglavju podamo sklepne ugotovitve in možnosti za nadaljnje delo.

1.3 Urotelij in rak sečnega mehurja

Urotelij je *prehodni epitelij* v ledvični kotanji, sečevodih, sečnem mehurju in proksimalni sečnici [3]. Glavna vloga sečnega mehurja je shranjevanje in periodično sproščanje urina. Študije prepustnosti kažejo, da je izmed vseh odkritih vrst epitelijev urotelij najmanj prepusten (transepitelijska upornost 10.000 do $> 75.000\ \Omega cm^2$; možganska ovojnica ima transepitelijsko upornost 1.000 do 2.000 Ωcm^2 [4]). Še vedno ni popolnoma raziskano, kako urotelij preprečuje prepuščanje vode, ionov, topljencev in strupenih agensov v kri in okoliška tkiva. Neprepustnost urotelija je ključna za normalno delovanje



Slika 1.1: Zgradba urotelija.

sesalcev, saj mora daljši čas zadrževati urin z vsebnostjo različnih stopenj sečnine, amoniaka in ostalih strupenih metabolitov. Ta lastnost po drugi strani postavlja omejitve za učinkovito absorpcijo zdravil. Ena izmed lastnosti, ki pripomorejo k neprepustnosti, je sferična oblika mehurja, ki minimizira razmerje površine urotelija in volumna shranjenega urina. Poleg tega je urotelij sestavljen iz več slojev različnih celic (slika 1.1). Na bazalno lamino in vezivno tkivo meji sloj majhnih bazalnih celic, nato sledi eden ali več slojev vmesnih urotelijskih celic. Sloj, ki meji na svetlino sečnega mehurja, je zgrajen iz visoko diferenciranih celic, ki jih včasih imenujemo tudi *dežnikaste celice*, saj jim njihova ploščata oblika omogoča, da kot dežnik pokrijejo več celic pod njimi. Urotelijske celice so znane po počasni fluktuaciji, pri miših življenjski cikel celic traja 40 tednov. V uroteliju obstajata dve poti prehodnosti: transcelularni in paracelularni. Pri transcelularni poti snovi prehajajo skozi plazmalemo celic, paracelularna pot pa poteka med celicami, je usmerjena in je lahko pasivna ali aktivna, nima določene smeri in sestoji iz difuzije in osmoze.[5]

Celice gredo med razvojem skozi vrsto genetskih in epigenetskih sprememb. Tekom tega procesa diferenciacije se izoblikujejo določene lastnosti, ki jim omogočajo opravljanje specializiranih nalog. Poškodbe ali podobni negativni vplivi lahko posledično pripeljejo do povečane proliferacije (celičnih delitev), ki omogoči zdravljenje oziroma popravilo poškodovanega tkiva. Pri nastajanju novotvorb se ta proces ne izvrši normalno, kar pripelje do nenadzorovanih celičnih delitev in oblikovanja tumorjev. Rakava obolenja lahko razumemo kot napako v postopku proliferacije, kot tudi diferenciacije celic. Izgubo diferenciacije lahko povežemo z zmanjšano zmožnostjo preprečevanja prepustnosti urotelija. Rakave celice kažejo številne izstopajoče lastnosti, med drugimi samozadostno rast, odpornost na inhibitorje rasti in neomejeno zmožnost celične delitve. Normalno diferencirane celice imajo namreč omejeno sposobnost celičnih delitev (končno število celičnih delitev). Tumorje ocenjujemo histološko glede na njihovo sestavo in značilnosti celic. Vse maligne novotvorbe urotelija kažejo vrsto pogostih histoloških značilk, povezanih s povečano proliferacijo in z izgubo zmožnosti diferenciacije.[2]

Mišično neinvazivna oblika raka sečnega mehurja (NMIBC - ang. non-muscle-invasive bladder cancer) je pogosta, heterogena bolezen, povezana z visoko stopnjo ponovljivosti. Možnosti zdravljenja so omejene in pogosto zahtevajo nadzor skozi vse življenje. Vključujejo transuretralno resekcijo rakavega tkiva, ki ji sledi kemoterapija ali imunoterapija. Vseživljenjski stroški zdravljenja na bolnika so med vsemi rakavimi obolenji najvišji. Bolniki namreč živijo dolgo, verjetnost ponovitve je velika, stroški nadzorovanja stanja pa visoki. Rak urotelija sečnega mehurja je pogosta oblika malignosti. Najpogostejši simptom je hematurija, ta se pojavi pri 80% do 90% bolnikih. V svetu NMIBC predstavlja približno 70% novo odkritih primerov raka sečnega mehurja. Pogosto prizadene starejše, mediana starosti bolnikov je 73 let. Večja incidenca je pri moških (3,81%) kot pri ženskah (1,18%) [1]. V Sloveniji je bilo v letu 2013 zabeleženih 344 novih primerov raka sečnega mehurja [6]. Predvidena incidenca v letu 2016 je 340 novih primerov (95%

interval zaupanja), od tega je 247 bolnikov moških [6]. Je 9. najpogostejša oblika raka pri moških in 15. najpogostejša pri ženskah; tveganje raka do 75. leta starosti je 0,8% (povprečje v Sloveniji v obdobju 2009-2013; [6]).

1.4 Pregled področja

Na področju raziskovanja rakavih obolenj je strojno učenje stalnica. Primarno se uporablja v diagnostiki in detekciji rakavih tvorb, v zadnjem času pa tudi vse bolj pogosto na področju prognoze. V pregledani literaturi smo našli veliko raziskav s področja obdelave slik in strojnega učenja na raku prostate in raku dojk [7]. Raziskave variirajo pri izbiri prostora značilk. Nekateri raziskovalci se tako osredotočajo na *makro* značilke (ponovna pojavitev raka, starost bolnika, ipd.), drugi na *mikro* značilke (sestava in morfologija rakavih tvorb).

V obdobju med letoma 1988 in 2016 je bilo v času pisanja objavljeno 585 člankov na temo strojnega učenja in analize morfologije celic (PubMed, ključne besede: *cancer, cancerous cells, cell detection, urothelial cancer, cell morphology, machine learning, data mining, supervised learning, image recognition, image segmentation, image analysis*).

Chen in sod. [8] v članku navajajo najprepoznavnejše napredke na tem področju, od odkrivanja rakavih celic do analize kromosomov in ostalih znotrajceličnih struktur. Avtorji navajajo, da se pri analizi mikroskopskih slik normalnih in rakavih celic pogosto uporablja metoda *Watershed*. Ta temelji na pristopu segmentacije z regijami, kjer si regije lahko predstavljamo kot poplavljena področja, če relief poplavimo z vodo. Pobočja in nakloni so pri računalniški analizi slik predstavljeni z različno intenzivnostjo sivin [9]. Drug algoritem za iskanje robov je *Canny edge*, ki robove zaznava na podlagi moči gradienta sivin - veliko razliko v intenzivnosti sivine zazna kot rob [10].

Med odprtokodnimi programi za analizo slik celic izstopa CellProfiler. Je modularen in omogoča analizo velikega nabora različnih dvodimenzionalnih slik, ni pa primeren za analizo daljših posnetkov in večdimenzionalnih slik. Orodje omogoča uporabo različnih izpeljav in implementacij metode Watershed ter ostalih metod za odkrivanje robov [11].

De Solórzano in sod. [12] ugotavljajo, da lahko s pomočjo analize fluorescenčnih slik izračunamo značilke, na podlagi katerih: 1) izmerimo prisotnost proteinov, 2) predvidimo njihovo znotrajcelično in zunajcelično razporeditev ter 3) izmerimo nestabilnost genomov v človeških in mišjih modelih raka dojke. V članku poudarjajo pomembnost velike učne populacije in prisotnosti poznavalca mikroskopiranja. Takšen pristop bi omogočil celostno pripravo baze podatkov o fenotipu raka, kar je ključno za razumevanje kdaj, kje in kako normalne celice postanejo rakave, kakor tudi za potencialno uporabo te baze podatkov v diagnostiki in nadaljnji terapiji. Glotsos in sod. [13] so razvili sistem za procesiranje slik, ki temelji na metodi podpornih vektorjev (SVM, Support vector machines) in služi vrednotenju stopnje napredovanja tumorjev na možganih. Algoritem je dosegel izjemno dobre rezultate (v povprečju 95% natančno odkrivanje jeder in 90,2% razločevanje med stopnjami tumorjev).

Tikkanen in sodelavci [14] predlagajo nov postopek za zaznavanje celic na svetlobno-mikroskopskih slikah. Celice na teh slikah so slabo kontrastne, kar oteži zaznavo celic. Opisana metoda se osredotoča na zaznavanje in štetje celic in ne na odkrivanje regij celic. Napovedni model uporablja metodo podpornih vektorjev s histogramom značilk usmerjenih gradientov. Vhod v koraku učenja so ročno označene slike. Zmogljivost metode je bila ovrednotena s 16 učnimi in 12 testnimi slikami, ki skupaj vsebujejo 10.736 celic raka na prostati. Avtorji navajajo visoko natančnost pri prečnem preverjanju z AUC nad 0,98 in povprečno relativno deviacijo 9% od ročno preštetih anotacij.

Wang in sodelavci [15] so razvili nov postopek za samodejno zaznavanje faze celičnega cikla. Segmentacija slik sestoji iz binarizacije in segmentacije z algoritmom Watershed. Vektor značilk vsebuje 211 značilk, med drugimi nekaj splošnih značilk, Haralick značilke, Zernikove momente in značilke pridobljene s transformacijo Gabor. Pred učenjem zmanjšajo prostor na 58 značilk. Napovedni model je zgrajen z metodo podpornih vektorjev. Klasifikator upošteva nove napačno uvrščene primere in se sproti posodablja. Avtorji navajajo dobre rezultate z visoko občutljivostjo in specifičnostjo. Trdijo, da metoda, ki jo predlagajo, pravilno prepozna 99% celičnih jeder. Med 18.683 jedri jih metoda 35 segmentira preveč, 154 pa premalo. Pri napovedovanju faze celičnega cikla rezultati nihajo. V zaključku avtorji poudarjajo pomembnost in vpliv pravilne segmentacije celičnih jeder na ostale korake sistemov za zaznavanje celic.

Hrebień in sodelavci [16] so v raziskavi na citopatoloških slikah raka na dojkah primerjali tri segmentacijske metode. Primerjava obsega metodo Watershed, aktivne konture in metodo GrowCut. Avtorji poudarjajo, da je formulacija problema težavna, saj je segmentacija domensko specifičen problem. Odkrivanje celičnih jeder je potekalo s strategijo evolucijskega (1+1) iskanja. Po besedah avtorjev segmentacija z algoritmom Watershed dosega 68,74%, metoda aktivnih kontur 22,32%, metoda GrowCut pa 10,4% ujemanje z ročno pripravljenimi predlogami.

Večina raziskav rokuje z vnaprej pripravljenim naborom podatkov. Čeprav je področje zelo dejavno, je v nekaterih raziskavah veliko pomanjkljivosti. Ponekod so učne množice slabo zastavljene, podatkov je premalo ali pa so nereprezentativni. Nekatere metode pri izgradnji učnih modelov so izbrane naivno; pogosto so uporabljeni modeli, ki so glede na velikost vzorca prezapleteni, zato so rezultati raziskav nezanesljivi. [7]

Poglavje 2

Metode in orodja

Namen poglavja je predstavitev metod in orodij, ki smo jih uporabili v implementaciji. Poglavje je razdeljeno na dva dela. V prvem delu govorimo o postopku segmentacije in obdelave slik, ki nam vrne označene celice, iz katerih lahko izračunamo značilke. Drugi del opiše koncepte, napovedne modele in orodja, ki na osnovi izračunanih značilk razločujejo med normalnimi in rakavimi urotelijskimi celicami.

2.1 Segmentacija in obdelava slik

Ta razdelek je namenjen opisu korakov samodejne segmentacije mikroskopskih slik. Označene regije so osnova za izračun značilk, ki jih potrebujemo za učenje napovednih modelov. Osrednji del postopka je algoritem Watershed, katerega implementacijo ponuja tudi program CellProfiler. Segmentacija je eden pomembnejših korakov pri samodejni obdelavi slik. Temelji na atributih slike, ki lahko predstavljajo sivino, barvo, teksturo, globino ali gibanje. Namen segmentacije je razbitje slike na smiselne regije v kontekstu problema, ki ga rešujemo. Sliko lahko segmentiramo na več načinov, najpogosteje uporabljene tehnike problem rešujejo z [9, 17] iskanjem robov, kjer z različnimi tehnikami iščemo razlike v intenzivnosti sivin (smer in moč gradienta), z gručenjem točk s podobnimi lastnostmi ali z rastjo regij.

Postopek segmentacije lahko grobo opišemo s sledečimi koraki:

- **zajem slik** z mikroskopom ali drugo napravo
- **pred-obdelava in odstranjevanje šuma:** za uspešno segmentacijo moramo odstraniti neželene atribute, ki se pojavijo kot stranski produkt zajema, saj ne sovpadajo z realno predstavo opazovanega objekta. Pri odstranjevanju šuma izgubimo del informacij o sliki. Primer takega šuma je neenakomerna osvetlitev pri zajemu slik.
- **odkrivanje regij:** sliko razbijemo na smiselne regije, ki predstavljajo opazovane objekte. Za vsako točko na sliki se najprej odločimo, ali pripada ozadju ali ospredju. Če je del ospredja, jo razvrstimo v regijo.

Segmentaciji sledi korak izbiranja značilk, kjer na osnovi odkritih regij izračunamo značilke, ki opisujejo opazovani objekt.

2.1.1 Algoritem Watershed

Algoritem *Watershed* je ena izmed najstarejših tehnik segmentacije. Prva sta ga predlagala Digabel in Lantuejoul. Beucher in Lantuejoul [10] v članku predlagata metodo za zaznavo obrisov, ki je neodvisna od parametrov (ne potrebuje določanja praga vrednosti). Osnovna ideja prihaja iz geografije, kjer nek topografski relief poplavimo z vodo, meje med posameznimi poplavljenimi področji pa predstavljajo robove med iskanimi segmenti. Pobočja in nakloni so pri računalniški analizi slik predstavljeni z različno intenzivnostjo sivin. Podobno si lahko predstavljamo, da so meje med regijami grebeni vrhov med dolinami, kjer se med padavinami dež izteče v ustrezno dolino. Lokalni minimumi predstavljajo točke, kjer se zbira deževnica, oziroma kjer regija začne svojo rast. [9, 10]

Prednost metode Watershed je, da vedno vrne zaprte obrise, kar je pri segmentaciji slik zelo pomembno. Primer metode, ki se uporablja pogosto, pa

tega ne zagotavlja, je metoda Canny Edge. Druga prednost metode Watershed je manjša računska zahtevnost v primerjavi z drugimi segmentacijskimi tehnikami. Njena slabost je občutljivost na lokalne minimume, kar lahko pripelje do nepravilne, prekomerne segmentacije. Med možnimi pristopi za reševanje problema prekomerne segmentacije so: [17]

- **združevanje regij** - po razbitju združimo manjše regije, ki pripadajo eni regiji,
- **delne diferencialne enačbe za odstranjevanje šuma ali poudarjanje robov** - pred razbitjem obdelamo sliko, se znebimo šuma in poudarimo strukture, ki delujejo kot ločnice med iskanimi regijami,
- **označevanje z markerji** - pred razbitjem primerno označimo točke na sliki, ki predstavljajo središča regij. S topološkega stališča so to lokalni minimumi ali lokalni maksimumi, ki jih izračunamo na podlagi gradienta ali katerih drugih atributov slike.

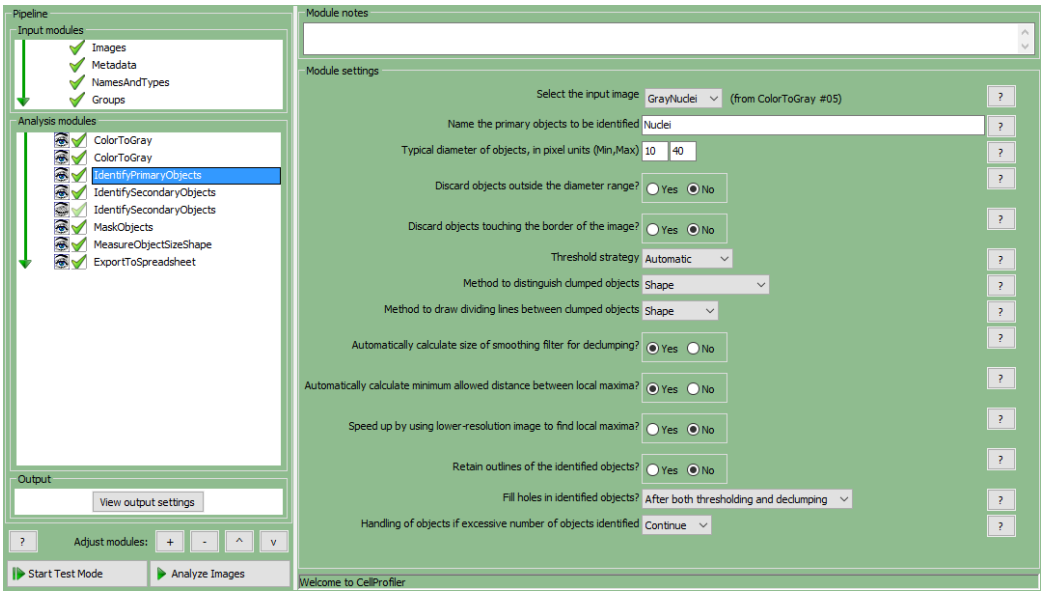
Ena večjih težav metode Watershed je posledica njene splošnosti. Veliko je variacij implementacij, ki se ne držijo standardnih definicij. Na rezultat, ki ga algoritem vrača, vplivajo tudi mnoge izboljšave hitrosti in prostorske zahtevnosti algoritma, kar pripelje do ne-determinističnega delovanja. To ni nujno opazno na vseh področjih aplikacije, je pa izredno pomembno pri analizi slik v medicini. [9]

2.1.2 CellProfiler

CellProfiler je prosto dostopen modularen programski paket za analizo slik, ki omogoča obdelavo serij tudi po več sto tisoč slik. Vsebuje implementacije metod za analizo različnih tipov celic. Je odprtokodna, prilagodljiva platforma za testiranje in razvoj novih metod. Program je razvit in optimiziran za delo z dvodimenzionalnimi slikami, podpira pa tudi analizo tridimenzionalnih slik in daljših časovnih posnetkov, ki pa je zelo omejena. Večina programske kode

je napisane v programskem jeziku MATLAB, ki je priljubljen v raziskovalnem svetu. Računsko zahtevnejše metode so implementirane v programskem jeziku C++. [11]

Kot lahko vidimo na sliki 2.1, celoten proces uporabe temelji na konceptu cevovoda. V svetu računalništva cevovod predstavlja verigo korakov (procesov, funkcij, rutin, ipd.), ki so urejeni tako, da je izhod posameznega koraka vhod v naslednji korak. Taka implementacija nam omogoča, da zasnujemo algoritem, kjer sami izberemo posamezne korake in njihov vrstni red. Algoritem lahko tako popolnoma prilagodimo problematiki področja. Na razpolago je velik nabor algoritmov, uporabnik pa lahko implementira tudi svoje module, ki jih nato na standarden način vključi v cevovod. Pomembno je poudariti, da programski paket ponuja nabor implementacij znanih in pogosto uporabljenih algoritmov. S tem se uporabniku prihrani trud z optimizacijo in pomisleke o pravilnosti implementacije, ki bi jo morebiti razvil sam. Osredotoči se na reševanje problema in ne na podrobnosti implementacije orodja.



Slika 2.1: Uporabniški vmesnik programa CellProfiler.

Uporabniški vmesnik je enostaven za uporabo in omogoča preprosto na-

stavljanje vrednosti različnih parametrov. Nabor parametrov se dinamično spreminja - odvisno od izbranega algoritma. Vse algoritme, ki so na voljo, spremlja podrobna dokumentacija. Ob ovrednotenju koraka v cevovodu, so uporabniku na razpolago v pregled vsi rezultati trenutnega koraka. Te lahko pred napredovanjem programa tudi izvozi. Izhode posameznih korakov in korake same lahko preprosto zadušimo ali onemogočimo.

2.1.3 Mahalanobisova razdalja

Kadar se soočamo s podatki multivariatne narave, je za izračun razdalje dobra izbira Mahalanobisova razdalja (MD). MD meri razdaljo med dano točko in središčem opazovane porazdelitve in je primerna tehnika za zaznavanje osamelcev v multivariatnih podatkih. [18].

MD pri dani naključni spremenljivki $U \sim N(\mu, \Sigma)$ je definirana kot: [19]

$$d(x, y) = ||T(x) - T(y)|| \tag{2.1}$$

kjer transformacija $T(x)$ slika U v standardno normalno porazdeljeno spremenljivko:

$$T(x) = \Sigma^{-\frac{1}{2}}(x - \mu)$$

Podobna definicija pravi, da je pri točkah $\vec{x} = [x_1, x_2, ..., x_p]^T$ in $\vec{y} = [y_1, y_2, ..., y_p]^T$ izbranih iz množice p spremenljivk s kovariančno matriko S dimenzij $p \times p$, Mahalanobisova razdalja d_m med točkami podana z:

$$d_m(\vec{x} - \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})} \tag{2.2}$$

2.1.4 Pragovni postopek z metodo Otsu

Postopek je razvil Nobuyuki Otsu [20] z idejo, da lahko pragovni postopek izpelje z vpeljavo kriterija, ki bi "uspešnost"praga vrednotil z bolj splošnega vidika, nato pa izbral njegovo optimalno vrednost. Metoda predvideva, da kot vhod zadostuje histogram vrednosti sivin (L možnih vrednosti) slike brez drugega predznanja. Formulacijo poenostavimo in histogram obravnavamo kot verjetnostno porazdelitev (enačba 2.3).

$$p_i = \frac{n_i}{N}, p_i > 0, \sum_{i=1}^{L} p_i = 1 \tag{2.3}$$

Slikovne pike glede na prag z vrednostjo k razporedimo v dva razreda C_0 in C_1, ki predstavljata ozadje in ospredje. Izpeljavo enačb za izračun verjetnosti (ω_0, ω_1), povprečnih vrednosti (μ_0, μ_1) in variance (σ_0^2, σ_1^2) razredov lahko bralec najde v prvotnem članku avtorja [20]. Da lahko ovrednotimo, kako dobra je določena vrednost praga, vpeljemo tri nove mere:

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}, \kappa = \frac{\sigma_T^2}{\sigma_W^2}\eta = \frac{\sigma_B^2}{\sigma_T^2}, \tag{2.4}$$

kjer so

$$\sigma_W^2 = \omega_0\sigma_0^2 + \omega_i\sigma_1^2 \tag{2.5}$$

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \tag{2.6}$$

in

$$\sigma_T^2 = \sum_{i=1}^{L}(i - \mu_T)^2 p_i \tag{2.7}$$

variance znotraj in med razredi ter varianca stopenj. Problem je tako omejen na optimizacijski problem, kjer iščemo vrednost praga k, ki maksimira eno izmed kriterijskih mer v enačbi 2.4. To stališče je rezultat domneve, da so razredi, ki jih pridobimo z dobrimi vrednostmi pragov, razdeljeni po vrednostih sivin. Posledično je najboljši prag tisti, ki najbolje loči razrede po vrednostih sivin.

$\sigma_W^2 + \sigma_B^2 = \sigma_T^2$ vedno drži, zato lahko rečemo, da je kriterij, ki ga optimiziramo, soroden kriteriju $\kappa = \lambda + 1$ in $\mu = \frac{\lambda}{\lambda+1}$. Opazimo, da sta σ_W^2 in σ_B^2 funkciji vrednosti praga k, σ_T^2 pa je od k neodvisen. Kot kriterijsko mero vzamemo μ, saj je od vseh treh najbolj preprosta (v odvisnosti od k). [20]

Optimalno vrednost k, ki maksimira μ, poiščemo s sekvenčnim iskanjem:

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k) \tag{2.8}$$

Interval iskanja lahko omejimo z:

$$S^* = \{k; \omega_0\omega_1 = \omega(k)[1 - \omega(k)] > 0, or 0 < \omega(k) < 1\} \tag{2.9}$$

2.2 Strojno učenje

V tem poglavju predstavljamo nekaj osnovnih konceptov in idej strojnega učenja ter odkrivanja vzorcev v podatkih. Opišemo vrsto orodij in klasifikacijskih napovednih modelov z različnih vej strojnega učenja. Podamo nekaj pogosto uporabljenih metrik, načina vrednotenja napovednih modelov in težav, s katerimi se pri tem soočamo.

Učenje lahko opišemo na sledeč način: [21]

> *Stvari se učijo, če spremenijo svoje obnašanje tako, da so v bodoče uspešnejše.*

Ta definicija postavlja uspešnost pred znanje. Nakazuje, da lahko učenje vrednotimo tako, da rezultate v danem času primerjamo z rezultati v neki točki v preteklosti. [21]

V grobem delimo algoritme strojnega učenja na dve vrsti:[22]

- **Nadzorovano učenje:** pri učenju iščemo primerno posplošitev podatkov - model, ki je lahko uporaben tudi za napovedovanje vrednosti prihodnjih podatkov. O razvrščanju oziroma klasifikaciji govorimo pri uvrščanju v diskretne razrede, pri zveznih ciljnih vrednostih pa o regresiji. V ta sklop spadajo metode, kot so odločitvena drevesa, naključni gozdovi, metoda najbližjih sosedov, nevronske mreže, ipd.

- **Nenadzorovano učenje:** za učenje uporabimo neoznačene primere. Algoritmu omogočimo, da iz množice primerov sam sklepa o porazdelitvi in sam odkrije možne strukture v podatkih. Med metode nenadzorovanega učenja spadajo gručenje (clustering), PCA, ICA, ipd.

V magistrskem delu smo uporabili nadzorovano učenje, saj smo poznali ciljne razrede, v katere smo želeli razvrščati. Preprost model, ki govori o algoritmih učenja je model PAC (Probably Approximately Correct). S teorijo o učenju konceptov iz primerov opisuje osnovno teorijo naučljivosti. Govori o razpoznavanju konceptov, ki v polinomskem času ločujejo med dvema razredoma (primer razredu pripada ali ne pripada). Sestavljen je iz protokola učenja, ki določa način pridobivanja informacij iz okolja, in postopka dedukcije, ki določa mehanizem, po katerem se izvede algoritem učenja. Za razumevanje modela sta ključna sledeča pojma:[21, 22]

- **Koncept** je to, kar se algoritem (neodvisno od vrste učenja) nauči. Je podmnožica celotnega prostora vseh možnih vzorcev in njihovih predstavitev. Izhod, ki ga model učenja ustvari, imenujemo opis koncepta. Cilj modela je, da se nauči neznan ciljni koncept $c \in C$. Proučujemo naučljivost konkretnega koncepta c, ki ga določa problem, in je eden izmed vseh možnih konceptov.

- **Vzorec ali primer** je posamezen, neodvisen primer koncepta, ki se ga učimo. Predstavimo ga kot vektor vrednosti atributov. Lahko so logične prireditve, realna števila, točke v Evklidskem prostoru, itd. Vsak atribut predstavlja eno značilko (feature) oziroma lastnost opazovanega objekta. Vhod učenja je torej tabela, katere vrstice imenujemo primeri, stolpce pa atributi oziroma značilke.

2.2.1 Ocenjevanje uspešnosti učenja

Med učenjem prilagajamo model našim podatkom. Iščemo torej tak model, ki najbolje opisuje porazdelitev vzorcev v učni množici. Če pri učenju

upoštevamo preveč parametrov, množica vzorcev pa je majhna, postane model prezapleten. Model se namesto pravih povezav med primeri priuči napak in šuma. Čeprav je lahko uspešnost na učni množici izvrstna, bo rezultat na novih primerih slab, saj se je model popolnoma prilagodil učnim podatkom. Dovolj velika množica vzorcev nam omogoči, da na tak šum v podatkih nismo tako občutljivi. Problematiko čezmernega prilaganja najdemo pri vseh algoritmih učenja. Metode reševanja so odvisne od izbranega algoritma in vključujejo izbiro testne množice, prečno preverjanje, regularizacijo in obrezovanje. [21, 22]

V praksi množico vzorcev pogosto razbijemo na učno in testno množico. Na prvi se učimo, drugo pa uporabimo pri vrednotenju algoritma po učenju. S tem preverjamo, ali se je to zgodilo in po potrebi zgradimo preprostejši model. Pri delitvi bi se lahko zgodi, da porazdelitev primerov po razredih bistveno odstopa od porazdelitve v celotni množici. To rešimo s stratificiranim vzorčenjem, ki primere izbira na tak način, da so porazdelitve razredov vseh vzorcih približno enake.

Boljši način izločanja pristranskosti je prečno preverjanje. Začnemo z vzorčenjem in razbitjem učne množice na N delov (npr. 10). Prvih $N - 1$ delov sestavlja novo učno množico, zadnji del pa novo testno množico. Sledi učenje, ki ga večkrat ponovimo, tako da vsakega izmed delov natančno enkrat uporabimo kot testno množico. Skupna ocena napake je povprečje posameznih napak, izračunanih med vrednotenjem na testni množici. Izkušnje kažejo, da je optimalna stopnja razbitja 10. Za boljše rezultate lahko prečno preverjanje kombiniramo s stratifikacijo, zaženemo večkrat in upoštevamo povprečje ocen napak.[21]

Napovedni model lahko na testni množici vrednotimo na več načinov. Kadar rešujemo problem, kjer napovedujemo, ali vzorec pripada razredu ali pa ne, govorimo o funkciji napake 0 - 1. Vmesnih možnosti ni; izguba je 0, če je napoved pravilna, ali 1, če je napoved napačna. Pogosto namesto tega uporabljamo verjetnostne napovedi razredov.[21]

Napovedna točnost (*classification accuracy*, CA)

Napovedno točnost izračunamo kot delež pravilnih napovedi. Ker ne upošteva verjetnosti, s katero razvrstimo posamezen primer v nek razred, ni zanesljiv pokazatelj zmogljivosti modela.

Krivulje ROC (*receiver operater characteristic*)

Krivulje ROC so grafična metoda za vrednotenje napovednih modelov. Prikazujejo zmogljivosti napovednega modela, brez upoštevanja porazdelitve razredov in stroška napake. Izrišejo razmerje deleža TP (2.10) na ordinatni osi in deleža FP (2.11) na abscisni osi. Resnično pozitivni primeri (TP) so tisti, kjer sta pravi in napovedani razred pozitivna, medtem ko je pri lažno pozitivnih primerih (FP) resnični razred negativen, naša napoved pa jih je uvrstila kot pozitivne. Poleg omenjenih poznamo še resnično negativne (TN) in lažno negativne (FN) primere. Bolj ko se krivulja približa levemu zgornjemu kotu, zmogljivejši je napovedni model, ki ga vrednotimo. Če se krivulja spusti pod diagonalo, opazujemo odziv, ki je slabši od naključnega vzorčenja. Delež resnično pozitivnih primerov (delež primerov, ki smo jih pravilno uvrstili kot pozitivne, izmed vseh pozitivnih primerov) imenujemo tudi *občutljivost*, delež resnično negativnih (delež primerov, ki smo jih pravilno uvrstili kot negativne, izmed vseh negativnih primerov) pa *specifičnost* (2.12).[21]

$$\text{sensitivity} = \frac{TP}{TP + FN} \tag{2.10}$$

$$FP_d = \frac{FP}{FP + TN} \tag{2.11}$$

$$TN_d = \frac{TN}{TN + FP} \tag{2.12}$$

Površina pod krivuljo ROC (*area under curve, AUC*)

Kadar želimo krivuljo ROC predstaviti s številom, uporabimo vrednost AUC (area under the curve), ki predstavlja površino pod krivuljo. V grobem lahko rečemo, da je model boljši, če je površina pod krivuljo večja. Površino pod krivuljo lahko tolmačimo tudi kot verjetnost, da model uvrsti naključno izbran pozitivni primer višje od naključno izbranega negativnega.[21]

2.2.2 Algoritmi učenja

Naivni Bayesov klasifikator

Naivni Bayesov klasifikator je preprost napovedni model, ki temelji na Bayesovem izreku pogojne verjetnosti (2.13).

$$P(c|x) = \frac{P(x|c) \times P(c)}{P(c)} \tag{2.13}$$

Metoda naivno predvideva, da so posamezne značilke medsebojno neodvisne, kar bistveno poenostavi pravilo (2.14).

$$\hat{y} = arg\max yP(y)\prod_{i=1}^{n} P(x_i|y) \tag{2.14}$$

Naivni Bayes je preprosta in hitra metoda, dobro se odreže tudi pri problemih z več ciljnimi razredi. Za učenje parametrov ne potrebujemo obsežnih množic vhodnih primerov. Kljub naivnosti se metoda dobro odreže tudi pri realnih problemih, čeprav se v praksi redko srečamo s problemom, kjer so značilke medsebojno popolnoma neodvisne. [21]

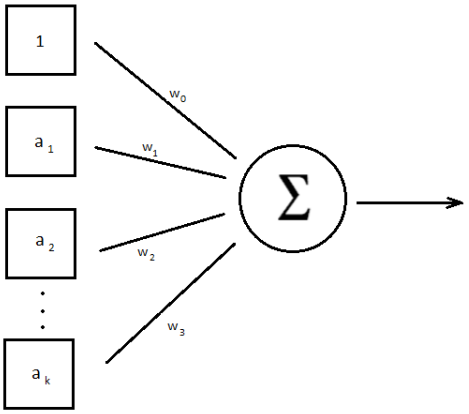
Naključni gozdovi

Naključni gozdovi so kombinacija odločitvenih dreves. Za k-to drevo izberemo naključni vektor Θ_k, ki je vzorčen neodvisno od prejšnjih naključnih vektorjev $\Theta, ..., \Theta_{k-1}$, a z isto porazdelitvijo. Drevo zgradimo tako, da uporabimo primere iz učne množice in vektor Θ_k. Dobimo klasifikator $h(x, \Theta_k)$, kjer je x vhodni vektor. Vzorec x razvrstimo v razred, za katerega glasuje

največ dreves. Generalizacijska napaka konvergira z večanjem števila dreves in je odvisna od zmogljivosti posameznih dreves ter korelacij med njimi. Stopnjo napake lahko zmanjšamo in naredimo robustnejšo, če pri delitvi vozlišč uporabimo naključno izbrane podmnožice značilk. [23]

Perceptron in nevronske mreže

Posamezen nevron ali perceptron je osnovni gradnik nevronske mreže. Predstavljamo si ga lahko kot graf (slika 2.2) z vozlišči in uteženimi povezavami. Sestoji iz dveh nivojev vozlišč: vhodni in izhodni nivo. Vhodni nivo ima eno vozlišče za vsak atribut, skupaj z dodatnim vozliščem, ki ima vedno vrednost 1, izhodni nivo pa vsebuje samo eno vozlišče. Vsako vozlišče na vhodnem nivoju je povezano z izhodnim nivojem, povezave med njimi so utežene.



Slika 2.2: Nevron - vrednosti a_i predstavljajo vhodne vrednosti značilk primerov, w_i uteži, Σ pa operacijo seštevanja.

Vhodni nivo se "aktivira", ko nevron na vhodu prejme nov primer. Vrednosti atributov se pomnožijo z utežmi, njihova vsota pa se posreduje na izhodni nivo. Izhodna vrednost je osnova za napoved ciljnega razreda. S stališča geometrije nevron razdeli prostor s hiper-ravnino. Nevron aktiviramo z vsemi primeri iz vhodne množice in sproti prilagajamo uteži na povezavah.

Kadar problem ni linearno razdvojljiv, ne bomo našli kombinacije uteži,

ki bi delovala na vseh primerih. V tem primeru več nevronov povežemo v hierarhično strukturo - nevronsko mrežo, kjer vmesne nivoje imenujemo "skriti nivoji". Poznamo več- in brez-nivojske nevronske mreže, take z in brez povezav nazaj, učimo jih lahko z nadzorovanim ali pa nenadzorovanim učenjem.[21, 22]

Metoda podpornih vektorjev (SVM)

Metoda podpornih vektorjev je kombinacija linearnega modeliranja in učenja na primerih, za katere je znano, kateremu razredu pripadajo. Model za vsak ciljni razred izbere majhno število mejnih primerov, ki jih imenujemo podporni vektorji. Na osnovi teh zgradimo linearne funkcije, ki te primere ločujejo tako, da je ločitvena meja čim širša. Linearno ločljivi problemi nam omogočajo, da iz odločitvene funkcije neposredno izpeljemo enačbo ravnine, ki ločuje primere. Metoda podpornih vektorjev torej poskuša maksimirati rob okrog te hiperravnine. Kadar problem ni linearno ločljiv, nam model omogoča, da uporabimo jedra višjega reda. Pri takšnem pristopu postanejo odločitve modela manj razumljive.[21, 22]

Ansambli odločitvenih modelov

Učenje, združevanje in kombiniranje več različnih ali enakih odločitvenih modelov se je izkazalo za dobro tehniko gradnje napovednih modelov. Najpogosteje uporabljene metode združevanja so bagging, boosting in stacking. Vsi trije modeli so v večini primerov zmogljivejši od posameznih odločitvenih modelov. Slabost takega združevanja je, da je kombinirane modele težko analizirati, njihove odločitve pa niso jasno razumljive. Kljub temu, da dosegajo dobre rezultate, ni jasno razumljivo, kateri faktorji in na kakšen način pripomorejo k boljšim napovedim.[21]

Bagging (bootstrap aggregating): Bagging je metoda kombiniranja napovednih modelov enakega tipa, kjer vsak model učimo na drugi naključno izbrani podmnožici primerov iste velikosti. Ciljni razred novega primera na-

povemo tako, da ga kot vhod podamo vsem napovednim modelom, nato pa izberemo tisti razred, ki se največkrat pojavi kot rezultat.[21]

Boosting: Podobno kot bagging, boosting združuje več napovednih modelov enakega tipa in uporablja glasovanje za odločitev o napovedanem razredu. Razlika med njima je v tem, da je boosting iterativen. Vsak nov model je odvisen od prejšnjih, saj pri njegovem učenju povečamo utež primerov, ki so jih prejšnji modeli napačno uvrstili. Napoved odločitvenega modela utežimo z napovedno točnostjo.[21]

2.2.3 Orange

Odločitvene modele smo gradili s programskim orodjema Orange. Orodje razvija Laboratorij za bioinformatiko, ki je del Fakultete za računalništvo in informatiko Univerze v Ljubljani. Ponuja programsko knjižnico implementirano v programskem jeziku Python. Med njegovimi prednostmi so preprosta in čista sintaksa, hitro učenje in programiranje ter preprosto razširljivost programskih modulov. Da bi uporabo programa omogočili večjemu razponu raziskovalcev, ki niso programerji, so razvijalci razvili uporabniški vmesnik, ki omogoča interaktivno obdelavo podatkov in gradnjo napovednih modelov v stilu cevovoda. [24]

Poglavje 3

Implementacija

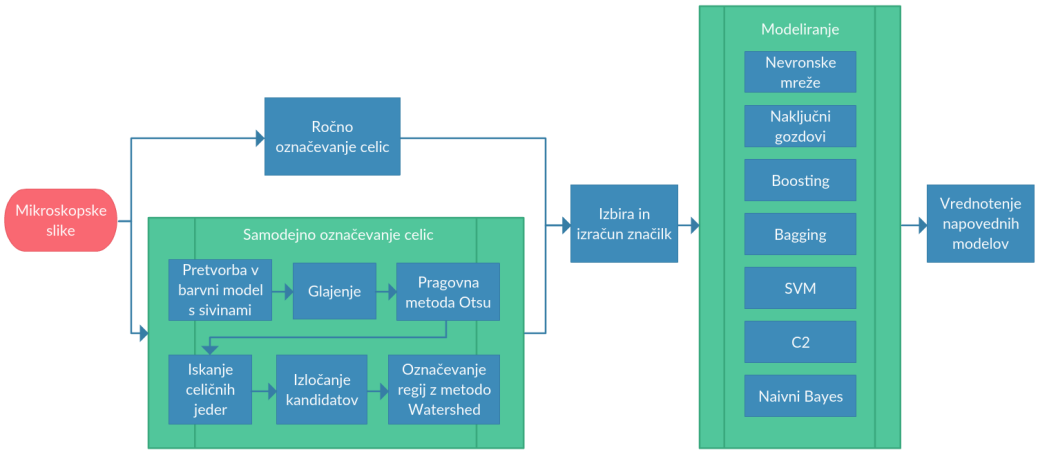
Celoten postopek je prikazan na sliki 3.1, razdelili smo ga na tri glavne sklope:

- **Obdelava mikroskopskih slik:** Zbirka mikroskopskih slik celic, ki jo je priskrbel Inštitut za biologijo celice na Medicinski fakulteti, Univerze v Ljubljani, vsebuje označene normalne in rakave urotelijske celice. Prvi korak je bil računalniško označevanje regij posameznih celic. V ta namen smo uporabili program CellProfiler [11], ki deluje na principu cevovoda. Glavna komponenta koraka je algoritem Watershed, s katerim smo segmentirali vhodne slike na podlagi odkritih celičnih jeder, celičnih membran in kontur površin celic (slika 3.2).

- **Izbor značilk:** Najdene celice so osnova za izračun značilk, kot so premer in velikost celice in jedra, celični indeks. Značilke smo izračunali z uporabo programskega paketa *MATLAB*, ki je olajšal delo s slikami in matričnimi operacijami.

- **Strojno učenje:** V zadnjem koraku smo izgradili in učili napovedni model, ki je napovedal, ali so celice rakave ali ne. Uporabili smo orodje Orange. Za izgradnjo modelov smo uporabili preprostejše univariatne oz. linearne metode, kot je naivni Bayesov klasifikator, kot tudi metode, ki zmorejo upoštevati tudi morebitne interakcije med značilkami, kot so naključni gozdovi in metoda podpornih vektorjev s primernimi

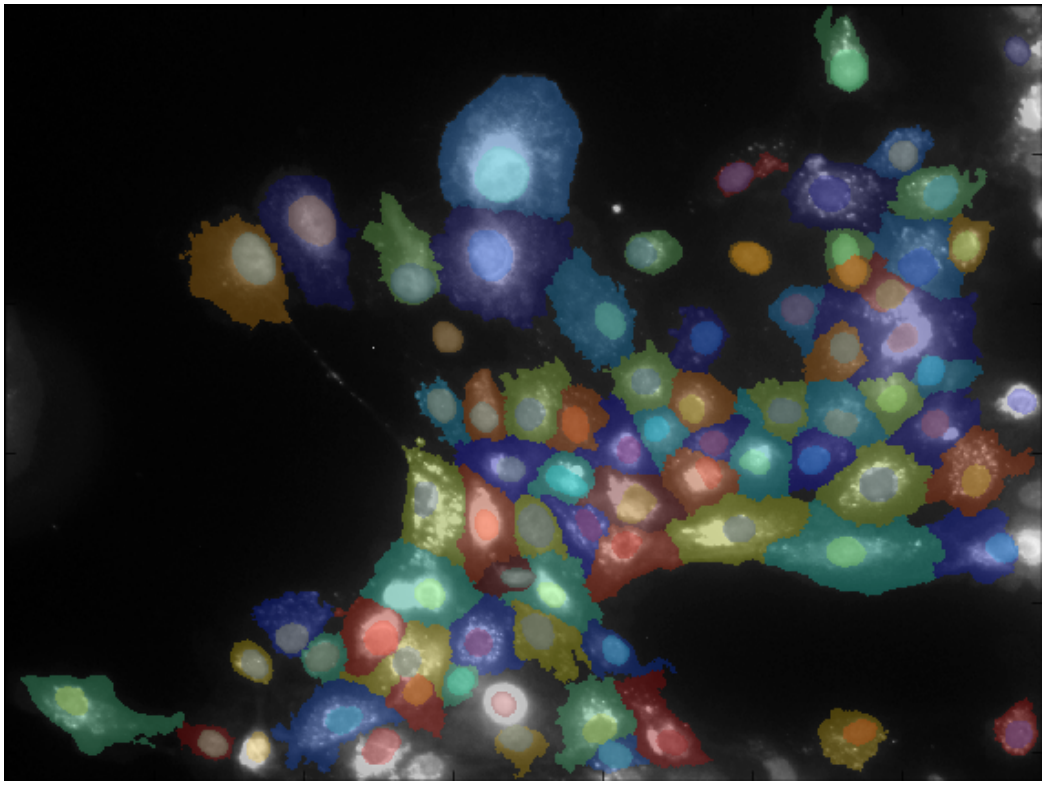
jedri. Ker se slike razlikujejo po zahtevnosti oziroma lahko vsebujejo specifične probleme, se je izkazalo za smiselno uporabiti tudi metode, kot sta AdaBoost in bagging (bootstrap aggregating).



Slika 3.1: Diagram, ki prikazuje postopek obdelave slik, ki mu sledi razvrščanje normalnih in rakavih urotelijskih celic.

Analizo mikroskopskih slik smo naredili tudi ročno (slika 3.3). Celice na slikah smo ročno označili s pomočjo uporabniškega vmesnika, ki smo ga implementirali v okolju MATLAB. Vmesnik omogoča označevanje (risanje) sklenjenih krivulj. Vsakič, ko celico označimo, program ponudi možnost, da celico razvrstimo kot rakavo ali normalno (empirično). Rezultati so osnova za izračunane referenčne značilke, ki služijo vrednotenju zaupanja v računalniško segmentacijo celic. Napovedni model smo skladno z običajno prakso vrednotili z učno in testno množico, kjer je vmesna evalvacija učenja na učni množici potekala s prečnim preverjanjem.

Sistem je v prvi fazi namenjen pomoči in podpori pri odločanju, tj. ne bo deloval avtonomno, temveč bo eksperta opozarjal na celice, ki jih mora (ročno) pregledati.

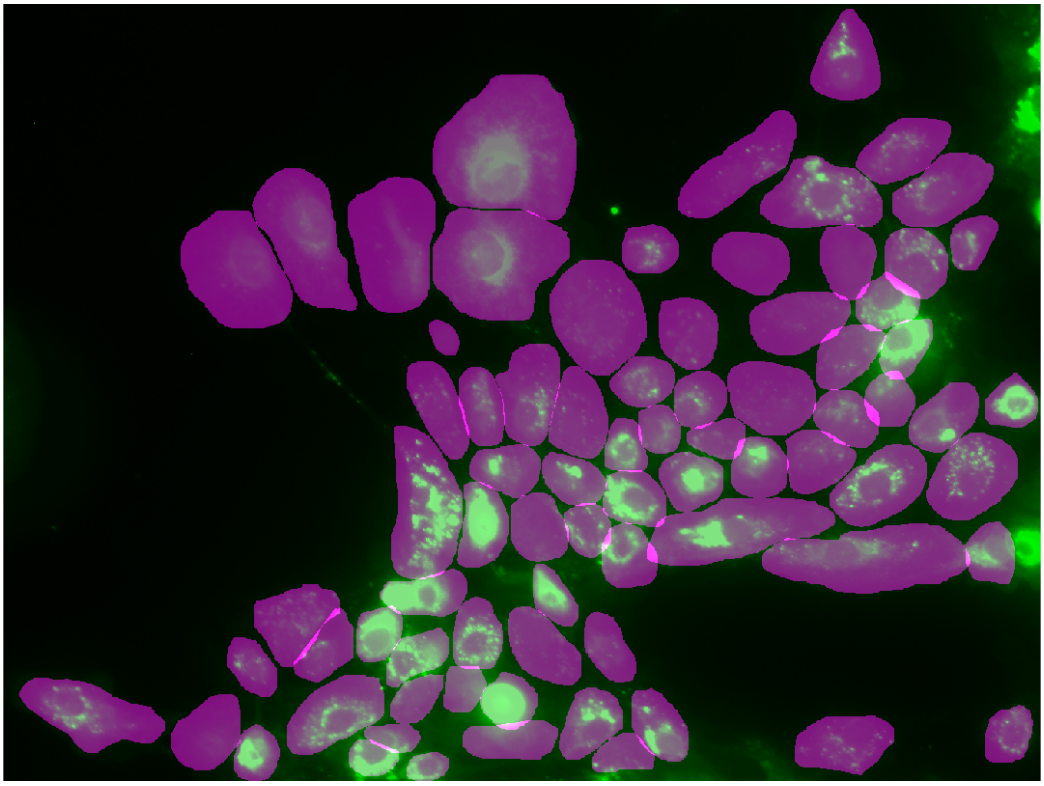


Slika 3.2: S samodejnim postopkom označene celice.

3.1 Podatki

3.1.1 In vitro modeli normalnih in rakavih urotelijskih celic

Podatke sestavlja množica 22 slik štirih različnih kokultur z različnimi nasaditvenimi gostotami. Gostota nasaditve je razvidna iz številke na koncu imena posamezne slike in priložene tabele 3.1. Primer mikroskopske slike je viden na sliki 3.4. Kokulture vsebujejo normalne prašičje in rakave humane urotelijske celice. Slike so bile posnete z objektivom z 20x ali 63x lastno povečavo na fluorescenčnem mikroskopu Zeiss AxioImager.Z1 z dodatkom Apoteme. Zapis informacij o celicah je shranjen v treh barvnih kanalih. Modra barva predstavlja DNA (jedra celic). Rdeča barva predstavlja fluorescentno označene rakave celice, medtem ko zelena barva označuje fluorescentno



Slika 3.3: Ročno označene celice.

označene normalne celice. Barvanje ni enako učinkovito pri vseh celicah. V nekaterih primerih se obarva le manjši del celične membrane, medtem ko pri nekaterih celicah opazimo, da barvilo šibko fluorescira tudi v citoplazmi celic. Barvilo se ponekod veže močneje kot drugje, kar se odraža v različnih intenzitetah obarvanosti. Tako na nekaterih mestih opazimo močno fluorescentne lise, drugje pa šibko obarvane regije. Ti pojavi otežijo segmentacijo in razločevanje tako človeku kot računalniku. Množico slik smo razdelili na učno in testno množico. Prva je služila učenju, druga pa vrednotenju modela, ki je razločeval med normalnimi in rakavimi urotelijskimi celicami.

3.1.2 Citopatološki urinski vzorci

V sodelovanju z Oddelkom za citopatologijo Onkološkega inštituta v Ljubljani smo obdelali 7 mikroskopskih slik urinskih vzorcev. Cilj analize je bil

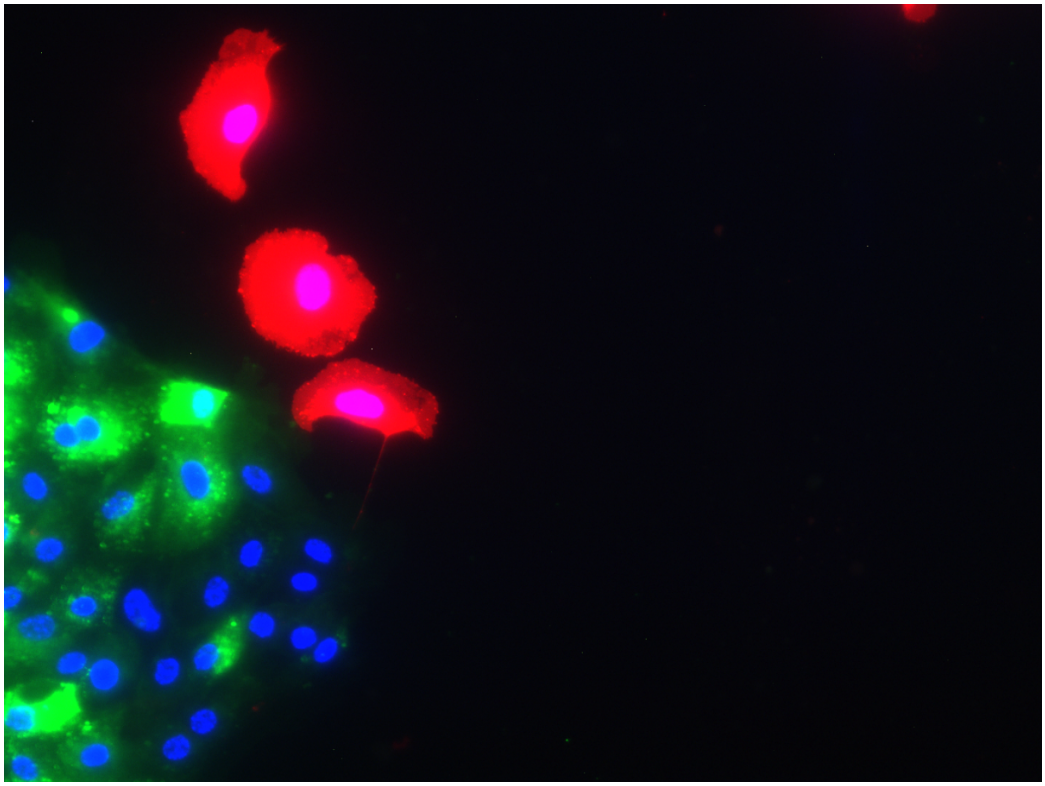
Oznaka kokulture	Nasaditvena gostota celic T24 (c/cm^2)	Nasaditvena gostota celic NPU (c/cm^2)
1	5×10^3	5×10^3
2	5×10^3	5×10^4
3	5×10^3	2×10^5
4	2×10^5	5×10^3

Tabela 3.1: Preslikava oznake kokulture in nasaditvene gostote.

preveriti delovanje algoritma in odločitvenega modela na slikah uporabljenih v diagnostičnih postopkih. Množici slik se med sabo razlikujeta, tako po tehniki slikanja, kot po vsebini. Ozadje slik s citopatološkega oddelka je obarvano belo, slike vsebujejo različne tipe celic z več slojev urotelija, kvaliteta slike variira v različnih področjih. Rakave celice se gručijo in so jasno razločne, prav tako hitro opazimo normalne diferencirane urotelijske (dežnikaste) celice. Velik delež vseh celic predstavljajo granulati, ki jih težko segmentiramo, saj so te celice majhne in slabo vidne. Pogosto opazimo, da se celice med sabo prekrivajo. V teh primerih s prostim očesom sklepamo o regiji, ki jo celica zavzema, samodejno prepoznavanje pa je pri tem koraku neuspešno. Jedra celic so vidno obarvana z modro barvo, ki je v kontekstu posamezne slike jasno razločna. Kljub temu nismo uspeli uporabiti filtra, ki bi zadovoljivo razločil jedra v vseh sedmih slikah. Razpoznavanje jeder je temeljni korak pri odkrivanju regij posameznih celic. Poleg celic so na slikah prisotni tudi drugi artefakti, ki niso del domene problema. Slika 3.5 prikazuje primer mikroskopske slike.

3.1.3 Barvanje celic

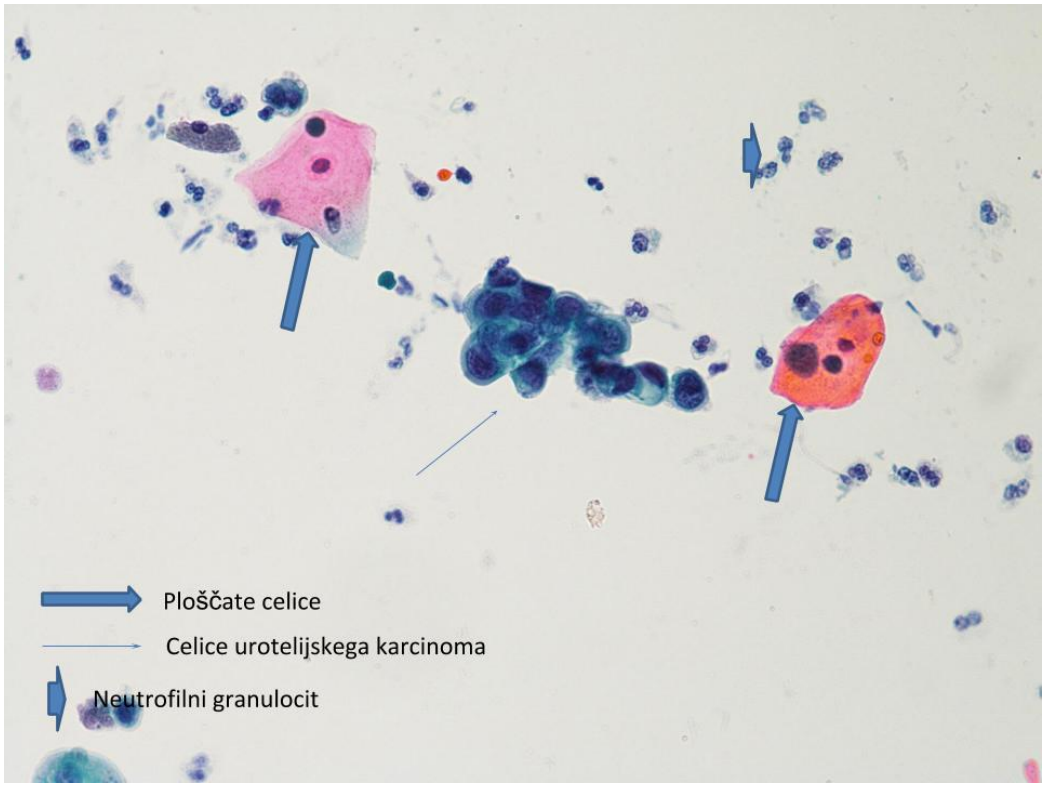
Mikroskopija omogoča opazovanje zelo majhnih struktur, ki so s prostim očesom nevidne. Slab kontrast nam včasih tudi s pomočjo mikroskopa otežuje razlikovanje različnih struktur v celicah. Temu v izogib pogosto uporabimo postopek barvanja celic. Namen barvanja celic je povečanje kontrasta z bar-



Slika 3.4: Primer fluorescenčne slike normalnih (zelene) in rakavih (rdeče) urotelijskih celic iz množice mikroskopskih slik. S prostim očesom razločimo razliko v velikosti in gručenju celic.

vanjem določenih opazovanih struktur, kar omogoči boljši pregled. Barvanje lahko poteka *in vivo* ali *in vitro*. Prva tehnika predvideva barvanje živih tkiv, pri tehniki *in vitro* pa barvamo tkiva, ki smo jih gojili v pogojih zunaj organizma. Pri računalniški obdelavi slik nam barvila olajšajo samodejno iskanje in označevanje struktur. Računalnik zlahka loči med različnimi barvnimi kanali. Za podrobnejše opazovanje je pogosta uporaba kombinacije več barvil. [25]

V danih podatkih so bila označena jedra celic in membrane. Uporabljena so bila barvila DAPI, DiI in DiO.



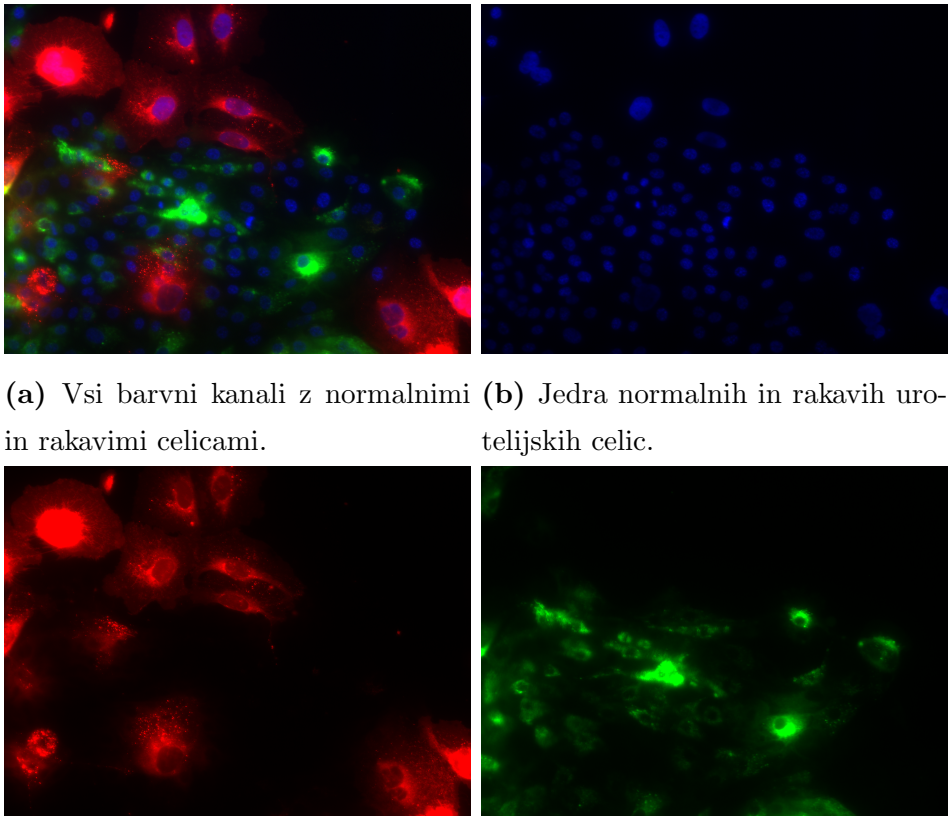
Slika 3.5: Mikroskopska slika celic patološkega urinskega vzorca.

- **DAPI** je fluorescentno modro barvilo, ki se veže na DNA in tako obarva celično jedro. Zaradi spektralnih značilnosti je zelo primeren za uporabo v kombinaciji z rdečimi in zelenimi barvili. Zaradi vezave na DNA se pogosto uporablja pri štetju celic, urejanju celic in kot orodje za segmentacijo pri obdelavi slik [26]. Slika 3.6b prikazuje modro obarvana jedra celic.

- **DiI** je fluorescentno lipofilno membransko barvilo fluorescentne oranžno-rdeče barve, ki obarva membrane celic. Pogosto se uporablja za dolgoročno označevanje živčnih in drugih celic [27]. Primer fluorescentno oranžno-rdeče obarvanih rakavih celic vidimo na sliki 3.6c.

- **DiO** je lipofilno fluorescentno barvilo zelene barve. V vodi je šibko fluorescentno, če ga vstavimo v membrano, pa močno fluorescira in je

stabilno na svetlobi. Znotraj membrane se razprši na vse strani. Prav tako obarva membrane celic [28]. S fluorescentno zeleno barvo obarvane normalne celice smo prikazali na sliki 3.6d.



(a) Vsi barvni kanali z normalnimi in rakavimi celicami.

(b) Jedra normalnih in rakavih urotelijskih celic.

(c) Barvilo DiI obarvalo rakave urotelijske celice.

(d) Barvilo DiO obarvalo normalne urotelijske celice.

Slika 3.6: Mikroskopske slike normalnih (zelene barve) in rakavih (rdeče barve) urotelijskih celic.

3.2 Računalniška obdelava slik

Človek hitro in z malo truda na slikah in v naravi prepoznava oblike in vzorce. Naši možgani zmorejo iz konteksta poudariti pomembne informacije in zanemariti ostale. Če za logično predstavitev ni dovolj podatkov, dopolnijo

sliko s tem, kar na tistem mestu pričakujejo. Zavedati se moramo, da so metode uporabljene pri računalniški obdelavi slik pogosto preproste in naivne. Uspešnost algoritmov je mnogokrat odvisna od prednastavljenih pragov, ki so pridobljeni empirično ali pa s strojnim učenjem. Uporaba strojnega učenja (tako nadzorovanega kot nenadzorovanega) pri računalniški obdelavi slik je zmožnosti računalnika pripeljala bližje človeškim.

Fleuret in sodelavci [29] so s preizkusi pokazali, da pri razvrščanju slik v razrede človek doseže boljše rezultate. Računalnik rezultat izboljša, če povečamo množico slik in obogatimo značilke uporabljene pri učenju. Za izgradnjo napovednega modela so uporabili metodi AdaBoost in metodo podpornih vektorjev. Borji in Itti [30] ugotavljata, da naloge, ki so računalniškim modelom težke, ljudje rešijo skoraj brez težav. Osredotočila sta se na problematiki prepoznavanja scen in prepoznavanja objektov. Primerjala sta 14 različnih modelov računalniškega vida na 7 naborih podatkov s 5 različnimi testi. Kljub temu težko rečemo, da je človek boljši pri obdelavi slik. Rezultati njunih testov kažejo, da je pri nekaterih nalogah računalnik uspešnejši. Razlika je še posebej očitna pri nalogah, kjer je človek uspešen, a prepočasen, kjer je vsebina slik neurejena in ni veliko globalnih informacij. Človek se v splošnem odreže bolje pri nalogah, ki zahtevajo prepoznavanje predmetov in scen v naravnih okoljih. Pri preprostih skicah visoke rezultate dosežemo že s klasičnimi modeli.

Računalnik lahko za specifične naloge z dobro zastavljenim učnim modelom in dovolj podatki izurimo hitro in dobro. Velike množice slik obdeluje hitreje in z manj napakami kot človek. Z napredovanjem računalniške obdelave slik je postalo rešljivih veliko problemov, ki so bili zaradi omejitev človeka nemogoči.

3.2.1 Segmentacija in odkrivanje regij

Označevanje regij, ki jih zavzemajo celice, smo opravili v dveh sklopih. V prvem delu smo uporabili program CellProfiler, nato pa smo vse celice označili še ročno. Opisani postopek segmentacije smo uporabili na množici slik, ki jih je priskrbel Inštitut za biologijo celice, na Medicinski fakulteti, Univerze v Ljubljani.

Lastna implementacija segmentacije celic

V prvem sklopu raziskave smo slike obdelali z algoritmom, ki smo ga implementirali v okolju MATLAB. Barvilo DAPI je obarvalo jedra celic z modro barvo 3.1.3. S pomočjo modrega kanala smo najprej označili jedra celic, te pa nato s kombinacijo erozije, dilacije in rekonstrukcije pretvorili v lokalne maksimume:

```
function [max] = localMaximums(I)
        se = strel('disk', 20);

        Ie = imerode(I, strel('disk', 13));
        Iobr = imreconstruct(Ie, I);
        Iobrd = imdilate(Iobr, se);
        Iobrcbr = imreconstruct(
                imcomplement(Iobrd),
                imcomplement(Iobr));
        Iobrcbr = imcomplement(Iobrcbr);

        max = imregionalmax(Iobrcbr);
end
```

Najdeni maksimumi so ključni za uspešno segmentacijo z algoritmom Watershed (poglavje 2.1.1). V naslednjem koraku smo iz slike izluščili ozadje

tako, da smo sešteli vrednosti vseh treh barvnih kanalov in zgradili masko, ki je pozitivna, kjer so vrednosti večje od nekega praga. Negativ maske predstavlja ozadje slike. Prvotno sliko smo iz barvne pretvorili v sivine in jo obdelali - okrepili smo kontrast in odstranili šum. Obdelano sliko smo uporabili kot vhod v algoritem Watershed, ki je sliko razbil na regije. Rezultatu smo odšteli ozadje in preostale strukture erodirali, da so ločnice med regijami postale bolj jasne. Na končni sliki smo z metodo *regionprops* izračunali središča in druge lastnosti regij, ter jih prikazali. Rdeče in zeleno barvilo sta služili referenčnemu označevanju normalnih in rakavih celic. Čeprav je ta postopek uspešno opravil segmentacijo posameznih slik, globalno ni vračal zadovoljivih rezultatov. Nekateri ključni koraki niso bili implementirani optimalno in so preveč odvisni od numeričnih parametrov, ki jih moramo nastaviti sami. Po empiričnem vrednotenju so bili rezultati kljub temu obetajoči, zato smo koncept s skoraj enakimi koraki preslikali v cevovod v programu CellProfiler.

Segmentacija s programom CellProfiler

Kot je opisano v razdelku 2.1.2, uporaba programa CellProfiler temelji na konceptu cevovoda. Koraki, ki smo jih vstavili v cevovod, so:

- **zajem slik:** vsak vhodni primer je sestavljen iz štirih barvnih slik v barvnem modelu RGB: prva vsebuje le modri kanal s celičnimi jedri, druga le rdeči kanal, kjer so obarvane rakave celice, tretja zeleni kanal z obarvanimi normalnimi celicami, četrta združuje vsebino vseh prvih treh slik.

- **označevanje slik:** vsaki sliki smo na vhodu podali oznako, ki opisuje njeno vsebino (jedra, rakave strukture, normalne strukture, celotna slika).

- **pretvorba v črno-bel barvni model**: slike iz barvnega modela RGB smo pretvorili v sivine različnih intenzitet.

- **odstranjevanje šuma - glajenje:** črnobelo sliko z jedri smo zgladili

z Gaussovim filtrom ($\sigma = 1$). Na ta način smo gladili robove struktur, ki se pojavljajo znotraj jeder celic.

- **odstranjevanje šuma - uporaba praga:** na zglajeni sliki jeder smo določili prag med ozadjem in ospredjem. Uporabili smo adaptivno metodo Otsu, ki točko uvrsti v ospredje ali ozadje tako, da minimizira entropijo znotraj vsakega razreda. Za velikost okna smo izbrali eno desetino velikosti slike. Z uporabo praga smo izločili predele s šibkimi intenzitetami sivin, ki bi jih algoritem pri odkrivanju regij lahko uvrstil v ospredje.

- **iskanje primarnih struktur**, ki so v nadaljnjih korakih služile kot izhodiščne točke pri iskanju površin celic (jedra celic).

- **izločanje kandidatov:** z empirično nastavljenim pragom smo odstranili neprimerne kandidate. Odstranili smo vse kandidate jeder, katerih premer je bil manjši od 15 slikovnih točk.

- **izvedba algoritma Watershed:** Na vhodu smo podali prvotno sliko, ki vsebuje vse strukture, in množico lokalnih maksimumov, ki smo jih izračunali v prejšnjem koraku.

- **izvoz regij:** izvozili smo datoteko v formatu .mat, ki vsebuje matriko. Format .mat omogoča preprosto branje in nalaganje v programskem okolju MATLAB. Matrika odraža lokacijo in površine regij na sliki. Vrednost točk, ki pripadajo regiji, so enake zaporedni številki regije. Koordinate posamezne točke so enake paru številk vrstice in stolpca v matriki. Ozadje je označeno z ničlami.

Na sliki 3.7 vidimo, kako uspešen je algoritem na različnih vrstah slik. Algoritem celice na sliki 3.7a segmentira zelo dobro, saj ta vsebuje velike, jasno razmejene celice na čistem ozadju. Čeprav so celice na sliki 3.7b bolj skupaj, je algoritem uspešen, saj so meje dovolj jasne. Manj uspešen je algoritem na sliki 3.7c - tam so nekatere celice majhne, se prekrivajo in gručijo,

nekatere so celo v procesu delitve.

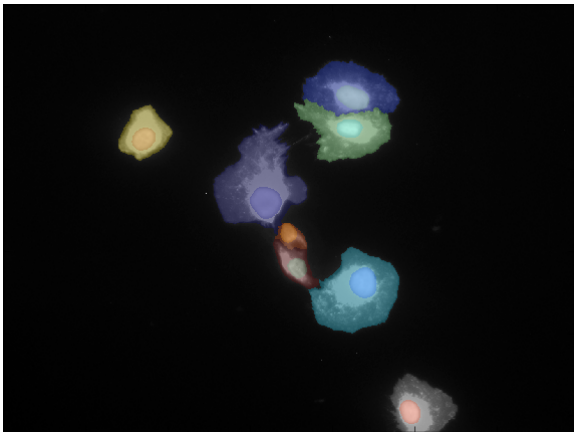
Ročno označevanje smo opravili pod nadzorom strokovnjaka, ki celice na ta način označuje v sklopu raziskovalnega dela na Inštitutu za biologijo celice. Namen tega koraka je bil oblikovanje referenčne množice vzorcev, s katerimi smo vrednotili rezultate samodejne računalniške segmentacije slik. Smiselna je primerjava števila odkritih celic in njihovih zabeleženih površin. Cilj je bil, da rezultat označevanja algoritmov približamo ročni segmentaciji. Poudarimo, da označbe strokovnjaka niso nujno pravilne. Izkaže se, da človek celično membrano oriše veliko bolj homogeno oziroma enakomerno, kot ta v resnici je. Čeprav ponekod že s prostim očesom opazimo nagubanost membrane, jo pri ročnem označevanju pogosto zanemarimo oziroma posplošimo. Debelina membrane celice je med 5 in 10 nm. Omejitev je tudi ločljivost svetlobnega mikroskopa (d = 200 nm).

3.2.2 Izbira značilk

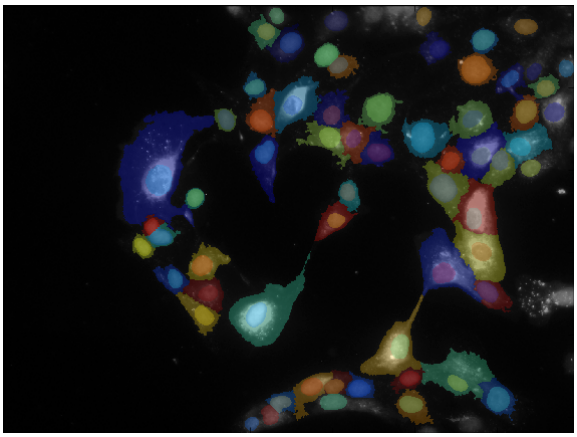
Vsako celico smo opisali z vektorjem 44 značilk in ciljnim razredom. Nekaj konceptov pomembnejših značilk smo podrobneje opisali v odstavkih nižje, vse značilke pa smo podali v tabeli 4.1. Postopek, ki izvede izračun značilk smo implementirali v okolju MATLAB. Funkcija za izračun na vhodu prejme ime slike in masko, s katero osamimo posamezno regijo. Kasneje smo dodali programski sloj, ki omogoča, da v kontekstu ene slike kot vhod podamo maske vseh odkritih regij. Z opazovanjem smo namreč ugotovili, da so nekatere značilke, s katerimi lahko empirično klasificiramo celice, odvisne tudi od soseske. S to izboljšavo smo nabor podatkov razširili z variacijo, kjer je vektor značilk utežen z vrednostmi značilk najbližjih sosed.

Metoda glavnih komponent (*Principal component analysis*, PCA)

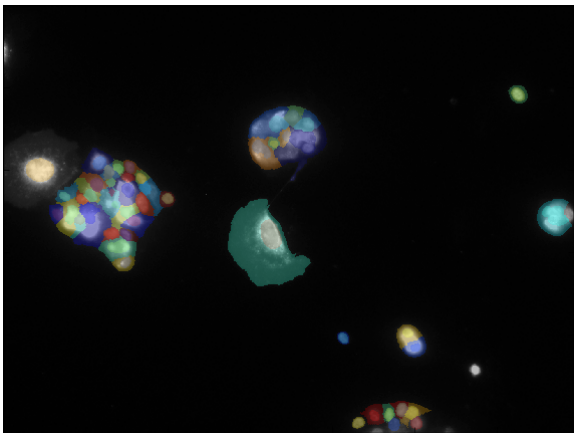
PCA (principal component analysis) ali metoda glavnih komponent je tehnika, s katero v podatkih odkrivamo vzorce. Cilj metode je, da zmanjšamo število dimenzij in odkrijemo tiste, ki najbolje opišejo dane podatke. Množico



(a) Primer dobro segmentirane slike.



(b) Primer dobro segmentirane slike.



(c) Primer slabše segmentirane slike.

Slika 3.7: Primerjava različno uspešne segmentacije slik.

spremenljivk, ki morda korelirajo, preslika v manjšo množico spremenljivk. Spremenljivke v končnem prostoru imenujemo *glavne komponente*. Metoda izvira s področja analize multivariatnih podatkov. Pogosto jo uporabimo kot prvi korak pri obdelavi ogromnih količin podatkov, odstranjevanju šuma in kompresiji podatkov. Metoda je sestavljena iz sledečih korakov:

- iz podatkov po vsaki dimenziji odštejemo njeno povprečje,
- izračunamo kovariančno matriko,
- izračunamo lastne vrednosti in lastne vektorje kovariančne matrike,
- izberemo število glavnih komponent,
- zgradimo nov nabor podatkov, tako da prvotne podatke pomnožimo z vektorjem značilk.

Glavne komponente so lastni vektorji kovariančne matrike, ki jih uredimo po velikosti lastnih vrednosti. Manjša ko je vrednost lastne vrednosti, manj informacij izgubimo, če se odločimo zavreči komponento. Prvih nekaj komponent vsebuje največ variance vseh prvotnih spremenljivk. Glavne komponente so med seboj ortogonalne.

Velikost in razmerja

Izsledki študij strokovnjakov na področju celične biologije in raziskav rakavih celic kažejo, da so morfološka razmerja dober kriterij pri prepoznavanju rakavih celic. V primerjavi z normalnimi celicami so rakave aktivnejše in večje po velikosti. Rakave celice so večje v splošnem, imajo pa tudi večja jedra. Omenjene lastnosti lahko izmerimo med računalniško analizo mikroskopskih slik in izkoristimo pri razločevanju s strojnim učenjem. Iz izkušenj je razvidno, da je pomemben pokazatelj tudi oblika elipsoide regije, ki jo celica zavzema. Opišemo jo lahko s kvocientom, ki ga imenujemo celični indeks.

Ena izmed značilk primerja dolžini najdaljših osi celice. Osi želimo poiskati neodvisno od spremenjenosti plašča oziroma nagubanosti membrane celice. Celico obravnavamo kot množico točk opazovane porazdelitve. Uporabimo PCA (poglavje 3.2.2), ta vrne glavne komponente, ki opišejo usmerjenost podatkov (množica točk, ki sestavljajo celico). Te so urejene po lastnih vrednostih, ki pripadajo lastnim vektorjem kovariančne matrike. Iskane vrednosti so razdalje od središča do membrane oziroma plašča celice, v smeri lastnih vektorjev. Rezultat je količnik dolžin najdenih osi.

Celični indeks (ICO)

Celični indeks ali indeks celične oblike (enačba 3.1) je merilo, ki nam grobo opiše obliko celice. Vrednost indeksa predstavlja razmerje med obsegom celice (najdaljši premer celice) in površino celice. Celice, katerih ICO je bližje vrednosti 1, so bolj kroglaste oblike, celice z ICO bližje 0 pa so bolj podolgovate oblike.

$$ICO = 4\pi \frac{povrsina}{perimeter^2} \tag{3.1}$$

Zernikovi momenti

Zernikove momente je v 30. letih prejšnjega stoletja vpeljal Fritz Zernike z namenom, da opiše optične anomalije. Kasneje so se ti atributi začeli uporabljati na področju obdelave slik, bolj podrobno pri prepoznavanju oblik. [31] Z matematičnega vidika so zanimivi zaradi svoje ortogonalnosti. Preprosto jih izračunamo s poljubno stopnjo in so rotacijsko invariantni. Višje stopnje vsebujejo več informacij o sliki, vendar so tudi bolj dovzetne za šum. [32]

Spremenjenost celične membrane

Spremenjenost celične membrane je pomemben dejavnik pri razločevanju med normalnimi in rakavimi celicami. Rakave celice so v primerjavi z normalnimi celicami veliko aktivnejše v okolju, kjer imajo dovolj prostora. Že na prvi

pogled lahko v odprtem okolju po velikosti celice s prostim očesom ločimo med rakavo in normalno celico. Aktivnost rakavih celic se v večini primerov kaže tudi v spremenjenem plašču, ki je nehomogene oblike s številnimi filopodiji in tunelskim membranskim nanocevkam podobnimi izrastki. Velikokrat lahko zasledimo primere, kjer rakava celica tvori nanocevke. To so cevaste strukture, ki jih rakava celica proži proti drugim normalnim in rakavim celicam v svoji soseski. Njihova primarna vloga je komunikacija in prenos snovi. [33]

Čeprav je spremenjenost oblike celice takoj prepoznavna s prostim očesom, to vseeno ni mera, ki jo preprosto ovrednotimo s samodejnim postopkom. Začnimo s predpostavko, da imamo podano in označeno regijo slike, ki jo zavzema opazovana celica. Naiven pristop k vrednotenju oblike je računanje standardnega odklona oddaljenosti posamezne točke membrane od središča celice z uporabo Evklidske razdalje. Izkaže se, da lahko dosežemo podobne vrednosti standardnega odklona pri podolgovati elipsoidi in liku, kjer sta glavni komponenti približno enako dolgi, točke na krožnici pa vseeno oscilirajo v svoji oddaljenosti od središča. Manj naivna in primernejša rešitev je uporaba **Mahalanobisove razdalje (MD)**, ki je opisana v razdelku 2.1.3.

Koeficient spremenjenosti oblike celice z MD smo izračunali na sledeč način: označili smo regijo slike, ki jo zavzema celica in jo pretvorili v množico dvodimenzionalnih točk. Množico točk smo razbili na množico, ki predstavlja krožnico oziroma plašč, in množico, ki predstavlja točke znotraj plašča. Za vsako točko na krožnici smo izračunali njeno Mahalanobisovo razdaljo do težišča točk porazdeljenih znotraj krožnice in izračunali standardni odklon izmerjenih razdalj.

Pomanjkljivost meritve ni v metodi sami, ampak podatkih, na katerih smo jo uporabili. Obrisi oziroma regije celic, ki smo jih dobili iz koraka računalniške segmentacije slik, ne odražajo realnega stanja v vseh primerih.

Za nek delež celic bo algoritem vrnil bolj homogene regije, kot so v resnici. Odstopanje od realnih oblik je še posebej očitno na ročno označenih slikah, saj je človeška roka manj natančna. Posledično so krožnice označene bolj enakomerno, kot bi jih označil algoritem.

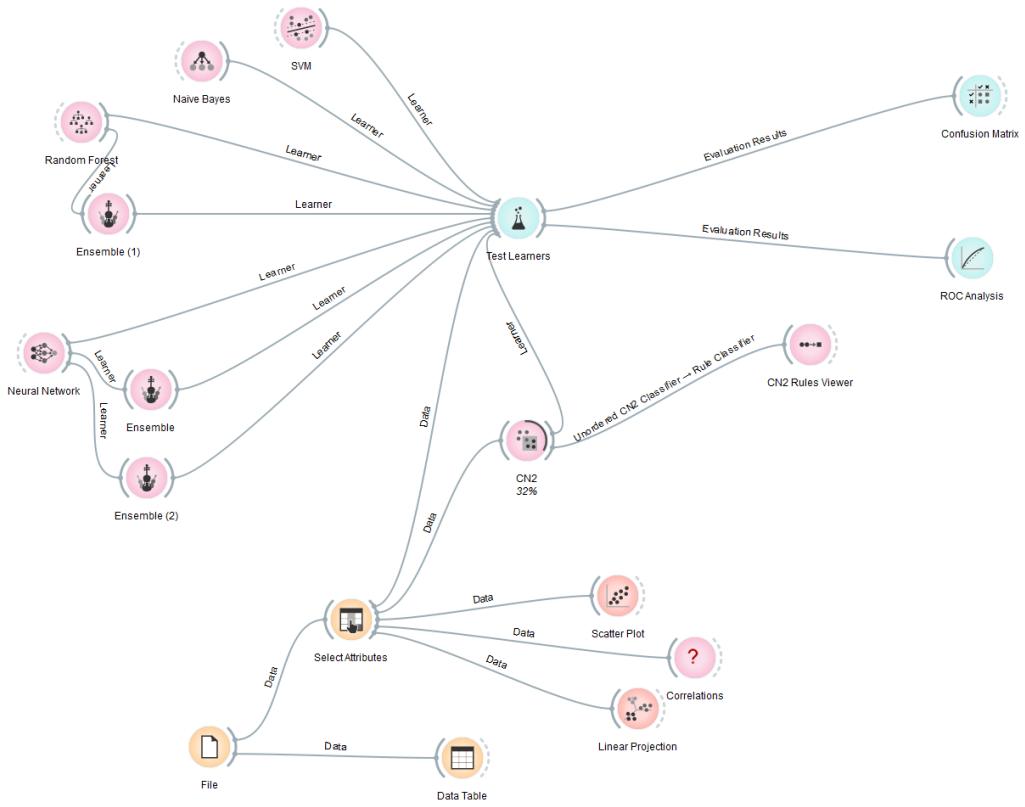
3.3 Modeliranje

Napovedne modele smo vrednotili s programom Orange [24]. Uporabili smo tri množice primerov. Prva množica vsebuje značilke ročno označenih celic. Druga množica vsebuje značilke primerov pridobljenih s programom CellProfiler in vključuje celice, ki s svojo površino mejijo na rob slike. Tretja množica je, podobno kot druga, sestavljena iz primerov pridobljenih s programom CellProfiler, z razliko, da smo tu izpustili celice, ki mejijo na rob slike. Napovedne modele smo v obeh orodjih vrednotili z 10-kratnim prečnim preverjanjem.

3.3.1 Orange

Program Orange razvija Laboratorij za bioinformatiko, Univerze v Ljubljani. Ponuja interaktivni uporabniški vmesnik, s katerim lahko grafično prikažemo korake obdelave podatkov.

Vsakemu gradniku je možno nastaviti vrsto parametrov. Prikazan proces (slika 3.8) smo začeli z gradnikom "File" in naložili datoteko s podatki, ki jih želimo uporabiti. Z gradnikom "Select attributes" smo izločili značilke, ki jih ne potrebujemo, in označili značilke, ki predstavljajo ciljni razred ali vsebujejo meta informacije o posameznem primeru. V tem koraku smo izločili značilko "density". Izločena značilka predstavlja gostoto poselitve celic na mikroskopski sliki. Izračunali smo jo na osnovi imena slike, iz katere izhaja opisana celica. Nekatere slike vsebujejo pretežno rakave ali pretežno normalne celice, tako ta značilka zelo dobro pripomore k razlikovanju v opazovani množici primerov. Tega podatka ne moremo uporabiti pri klasifikaciji



Slika 3.8: Procesni graf, ustvarjen s programom Orange.

novih slik. Statistične soodvisnosti pridobljene množice primerov smo izmerili in si jih ogledali z gradniki "Correlations", "Scatter Plot" in "Linear Projection". Gradnik "Test learners"je osrednje vozlišče našega grafa. Njegova naloga je, da oceni zmogljivost napovedovanja napovednih modelov na dani množici primerov. Na vhodu sprejme predpripravljeno množico primerov in nabor napovednih modelov, na izhodu pa vrne rezultate ovrednotenja.

V tabeli 3.2 smo podali izbrane napovedne modele in parametre učenja. Modele smo vrednotili s prečnim preverjanjem, parametre modelov pa izbrali z notranjim prečnim preverjanjem.

model	parameter	vrednost
SVM	meja kompleksnosti v	0,5
	g	0,11
	tip	v-SVM
	jedro	RBF $(exp(-g\|x-y\|^2))$
	numerična toleranca	0,0010
nevronske mreže	velikost skritega nivoja	50
	stopnja regularizacije	1,1
	maks. št. iteracij	400
naivni Bayes	metoda ocenjevanja	relativna frekvenca
naključni gozdovi	št.dreves	100
	maks. globina delitve	5
boosting	gnezden model	odločitveno drevo
	št. instanc	10
	min. št. primerov v drevesu	2
	kriterij izbire atributov v drevesu	informacijski dobitek
bagging	gnezden model	odločitveno drevo
	št. instanc	10
	min. št. primerov v drevesu	2
	kriterij izbire atributov v drevesu	informacijski dobitek
CN2		

Tabela 3.2: Izbrani modeli in parametri učenja.

Poglavje 4

Rezultati in vrednotenje

Množica primerov ročno označenih celic nosi oznako R. Primeri, ki smo jih pridobili s programom CellProfiler, so označeni z oznako CP1. Oznaka CP2 označuje primere pridobljene, podobno kot CP1, s programom CellProfiler, le da ne vsebuje celic, ki mejijo na rob slike. Krivulje ROC (slike 4.1, 4.2 in 4.3) lepo prikažejo razlike med izbranimi napovednimi modeli. Že na prvi pogled je razvidno, da se je metoda naivni Bayes odrezala najslabše (slika 4.1). Najbolje je razločevala metoda nevronske mreže (AUC 0,9052), tesno ji sledita metodi bagging z odločitvenimi drevesi (AUC 0,9041) in naključni gozdovi (AUC 0,9005).

#	Značilka	ReliefF	Inf. Gain	Gain Ratio	Gini
1	stats_orientation	0,067	0,004	0,002	0,001
2	stats_minorAxis	0,51	0,238	0,119	0,071
3	stats_area	0,042	0,165	0,082	0,050
4	zernike_1_1	0,041	0,169	0,084	0,052
5	stats_equivDiameter	0,040	0,165	0,082	0,050
6	corePercent	0,038	0,034	0,018	0,011
7	coreArea	0,036	0,183	0,091	0,057
8	stats_eccentricity	0,036	0,048	0,024	0,015
9	zernike_6_2	0,033	0,044	0,022	0,014

10	stats_perimeter	0,032	0,119	0,059	0,037
11	avg_corePercent	0,027	0,038	0,019	0,012
12	zernike_3_1	0,026	0,060	0,030	0,019
13	zernike_7_3	0,026	0,037	0,018	0,012
14	shapeSD	0,024	0,128	0,064	0,040
15	zernike_7_7	0,023	0,030	0,015	0,009
16	zernike_6_6	0,021	0,026	0,013	0,008
17	avg_stats_eccentricity	0,020	0,009	0,004	0,003
18	avg_coreArea	0,020	0,013	0,006	0,004
19	avg_stats_extent	0,017	0,006	0,003	0,002
20	zernike_3_3	0,017	0,023	0,011	0,007
21	stats_majorAxis	0,016	0,070	0,035	0,022
22	zernike_2_2	0,014	0,027	0,013	0,009
23	stats_extent	0,013	0,019	0,009	0,006
24	zernike_4_4	0,013	0,028	0,014	0,009
25	zernike_8_2	0,013	0,054	0,027	0,017
26	avg_stats_minorAxis	0,013	0,010	0,005	0,003
27	zernike_8_8	0,013	0,038	0,019	0,012
28	zernike_7_5	0,012	0,033	0,016	0,011
29	shapeMahal	0,010	0,002	0,001	0,001
30	zernike_4_2	0,010	0,033	0,016	0,010
31	zernike_8_4	0,010	0,041	0,020	0,013
32	avg_shapeSD	0,009	0,008	0,004	0,002
33	zernike_7_1	0,009	0,042	0,021	0,013
34	zernike_6_4	0,008	0,039	0,019	0,013
35	zernike_5_3	0,007	0,031	0,015	0,010
36	avg_stats_perimeter	0,006	0,008	0,004	0,002
37	avg_stats_equivDiameter	0,005	0,015	0,007	0,005
38	avg_stats_majorAxis	0,005	0,004	0,002	0,001
39	zernike_8_6	0,004	0,037	0,018	0,012
40	avg_shapeMahal	0,004	0,000	0,000	6.619e-05

41	avg_stats_area	0,003	0,010	0,005	0,003
42	zernike_5_5	0,002	0,023	0,011	0,007
43	avg_stats_orientation	-2.92e-05	0,011	0,005	0,003
44	zernike_5_1	-0,000	0,073	0,036	0,023

Tabela 4.1: Značilke, ki smo jih uporabili pri učenju.

V tabeli 4.1 smo prikazali značilke razvrščene po vrednosti metrike ReliefF. Poleg smo zabeležili še informacijski prispevek (*Inf. Gain*), relativni informacijski prispevek (*Gain ratio*) in vrednost indeksa Gini. Vidimo, da s ciljnim razredom ni korelirala izrazito nobena značilka. Najvišjo vrednost ReliefF je imela značilka, ki beleži orientacijo elipsoide celice (*stats_orientation*, vrednost ReliefF 0,067). Razvrstitev po vrednosti informacijskega prispevka (0,238), relativnega informacijskega prispevka (0,119) in indeksa Gini (0,072) izpostavi značilko *stats_minorAxis*. Ta predstavlja dolžino krajše izmed obeh osi elipsoide celice. Nekaj parov značilk, ki so najbolj korelirali, smo prikazali v tabeli 4.2. Značilka *shapeMahal* predstavlja standardni odklon Mahalanobisove razdalje, *stats_[minor/major]Axis* beležita dolžini osi elipsoide celice, *stats_area* predstavlja površino regije celice, *stats_perimeter* pa obseg celice. Soodvisnost opazovanih značilk je pričakovana, saj vse opisujejo obliko celice.

Kot vidimo v tabeli 4.3 je prvi nabor podatkov z oznako R opazno manjši od tistega z oznako CP1. Razlogov za to je več: program CellProfiler je ne-

značilka 1	značilka 2	Pearson	Spearman
stats_majorAxis	stats_area	0,8720	0,9183
stats_minorAxis	stats_area	0,9216	0,9219
stats_perimeter	stats_majorAxis	0,9630	0,9717
stats_area	shapeMahal	0,8526	0,8832

Tabela 4.2: Pari značilk z najvišjimi korelacijami.

oznaka	opis	št.	št. rakavih	št. normalnih
R	ročno označeni	1224	467	757
CP1	CellProfiler	1653	586	1067
CP2	CellProfiler, brez mejnih celic	1460	513	947

Tabela 4.3: Število primerov in porazdelitev ciljnega razreda v posameznih množicah podatkov.

katere regije prekomerno segmentiral, rezultat tega je več odkritih segmentov, kot jih v resnici je; razlog bolj subjektivne narave najdemo v odločitvi označevalca, da izpusti celice, ki mejijo na rob slike ali pa niso jasno vidne. Množica primerov CP2 je po pričakovanjih manjša od množice CP1, saj ne vsebuje celic, ki se dotikajo roba slike.

Učinkovitost algoritma za segmentacijo mikroskopskih slik smo vrednotili empirično. Algoritem je zelo dobro zaznal in označil celice z jasnimi mejami in teksturami. Slabše se je odrezal pri skupinah celic, ki so v procesu delitve, so tesno skupaj ali se prekrivajo. Človek pomisli tudi na to, da imajo rakave celice proliferacijo in da so celice v delitvi po vsej verjetnosti rakave ali nedeljene bazalne/vmesne urotelijske celice. V teh primerih je algoritem pogosto segmentiral prekomerno, predvsem pri celicah v delitvi, kjer je zaznal več jeder in posledično predvidel več celic. Pomembno je poudariti, da so v takih situacijah neodločni tudi strokovnjaki, ki celice označujejo ročno. Ker se strokovnjaki odločajo na podlagi barvil, v teh primerih točnost označevanja ni tako pomembna, kot pri samodejnem zaznavanju. V tej raziskavi smo ročno označene celice obravnavali kot referenčno množico vzorcev. Zavedamo se, da so označbe strokovnjakov mnogokrat približne in ne sovpadajo vedno z resnično obliko celice. To v podatke vnese šum in lahko negativno vpliva na odkrivanje vzorcev in povezav med značilkami. Na sliki 4.4 vidimo tri različne stopnje segmentacije slike. Sivine (4.4a) so osnova za rast regij z algoritmom Watershed, druga stopnja (4.4b) vsebuje označene membrane celic, končni rezultat segmentacije (4.4c) pa označene celotne površine odkritih regij.

Rezultate vrednotenja napovednih modelov smo podali v tabelah 4.4 in 4.5. Vsi izbrani napovedni modeli so dosegli dobre rezultate na danih podatkovnih množicah. Najboljše rezultate so pričakovano dosegli z množico primerov z ročno označenimi celicami. Omenili smo, da množica podatkov CP2 ne vsebuje celic, ki mejijo na rob slike. Vidimo le del teh celic, zato so vse meritve, ki smo jih opravili na teh celicah, neuporabne in so v podatke vpeljale šum. Pričakovali bi, da je napovedna uspešnost pri množici podatkov CP2 boljša kot pri CP1, pa temu ni tako. Napovedna točnost in vrednost AUC sta za bili nekatere napovedne modele višji pri množici podatkov CP1, pri nekaterih pa za CP2. Odstopanja napovedne točnosti in vrednosti AUC so majhne, kar kaže na to, da izločitev robnih celic ni bistveno pripomoglo k uspešnosti razločevanja. V tabeli z rezultati (4.4, 4.5) vidimo, da so med izbranimi metodami najboljše rezultate dosegle metode bagging, nevronske mreže in naključni gozdovi, sledi metoda SVM. Najvišjo vrednost AUC so na množici primerov R dosegle nevronske mreže (0,9052), tesno sledita metodi bagging (0,9041) in naključni gozdovi (0,9005).

V zadnjem delu smo napovedovanje opravili še na množici ročno označenih slik (U) urinskih vzorcev, ki smo jih pridobili z Onkološkega inštituta v Ljubljani. Ker so bile slike ustvarjene z drugačnim postopkom mikroskopiranja, niso primerne za samodejno segmentacijo z našim postopkom. Vsebujejo množico različnih tipov celic in drugih struktur, ki se jih med ročnim označevanjem izpusti. Napovedne modele smo učili na zbirki podatkov R, nato pa kot testno množico primerov uporabili množico U. Rezultati kažejo (tabela 4.6), da se najbolje odrežejo naključni gozdovi (AUC 0,8778) in nevronske mreže (AUC 0,8634). Metoda bagging, ki je bila pri ostalih množicah vzorcev boljša, je dosegla slab rezultat (AUC 0,6882).

Podatki	Napovedni model	AUC	Natančnost
R	SVM	0,8935	0,8291
	naključni gozdovi	0,9005	0,8393
	naivni Bayes	0,7762	0,7239
	nevronske mreže	0,9052	0,8575
	CN2	0,8337	0,7919
	boosting	0,7725	0,8211
	bagging	0,9041	0,8391
CP1	SVM	0,8392	0,7997
	naključni gozdovi	0,8456	0,7913
	naivni Bayes	0,6930	0,7644
	nevronske mreže	0,8566	0,8364
	CN2	0,7722	0,7700
	boosting	0,7121	0,7970
	bagging	0,8475	0,8030
CP2	SVM	0,8361	0,8015
	naključni gozdovi	0,8556	0,8104
	naivni Bayes	0,7118	0,7817
	nevronske mreže	0,8485	0,8279
	CN2	0,7900	0,7957
	boosting	0,7004	0,7867
	bagging	0,8478	0,8057

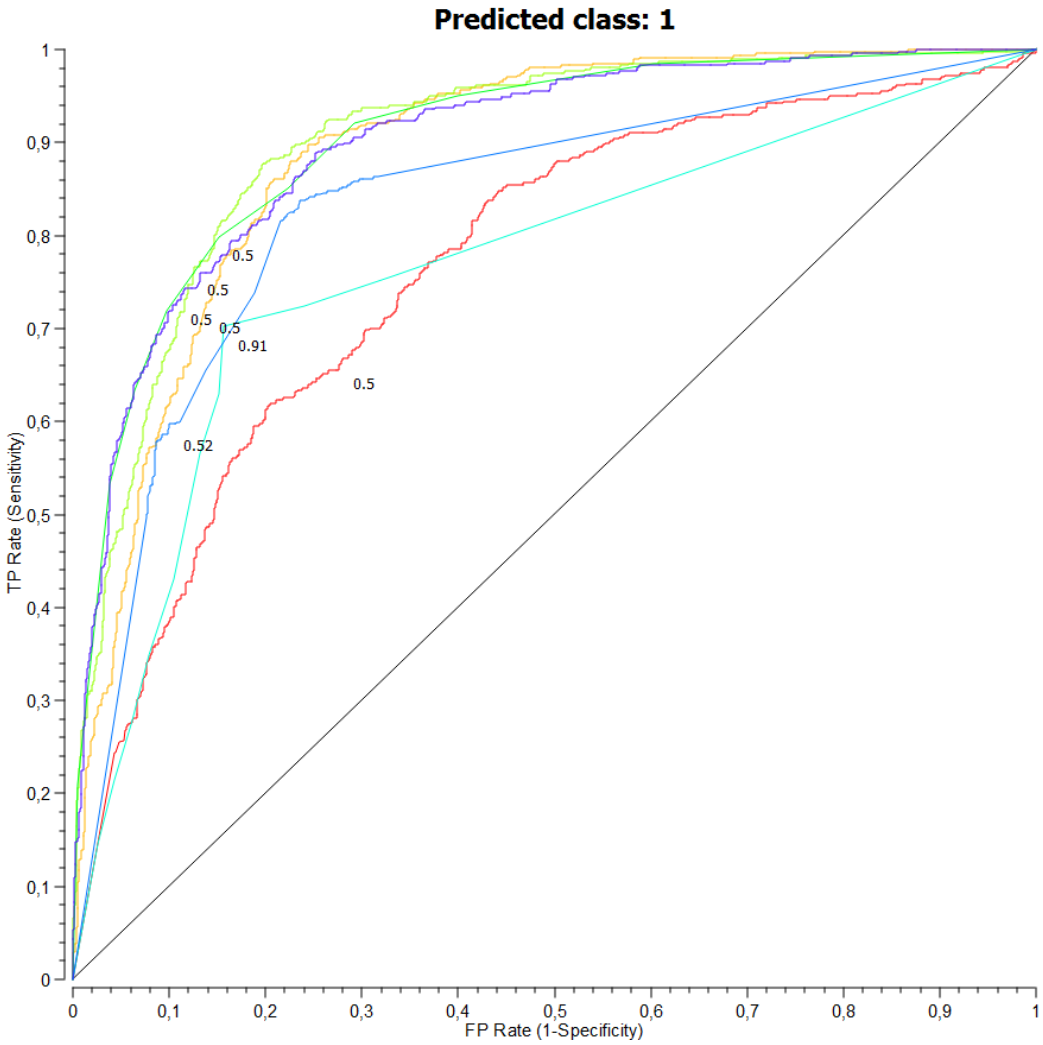
Tabela 4.4: Vrednosti AUC in natančnost napovednih modelov. R - ročno označene celice; CP1 - celice označene s programom CellProfiler; CP2 - celice označene s programom CellProfiler, brez mejnih celic.

Podatki	Napovedni model	CA	Specifičnost	Občutljivost
R	SVM	0,8064	0,7109	0,8626
	naključni gozdovi	0,8333	0,7195	0,9036
	naivni Bayes	0,7002	0,6617	0,7239
	nevronske mreže	0,8244	0,7687	0,8653
	CN2	0,7941	0,6146	0,9049
	boosting	0,7892	0,7024	0,8428
	bagging	0,8325	0,7195	0,9022
CP1	SVM	0,7961	0,5836	0,9128
	naključni gozdovi	0,7834	0,5666	0,9025
	naivni Bayes	0,6600	0,6160	0,6842
	nevronske mreže	0,8125	0,6860	0,8819
	CN2	0,7592	0,5137	0,8941
	boosting	0,7350	0,6331	0,7910
	bagging	0,7973	0,5939	0,9091
CP2	SVM	0,7986	0,5809	0,9166
	naključni gozdovi	0,8000	0,6101	0,9029
	naivni Bayes	0,6589	0,6608	0,6579
	nevronske mreže	0,8082	0,6589	0,8891
	CN2	0,7897	0,5692	0,9092
	boosting	0,7308	0,5984	0,8025
	bagging	0,8062	0,5887	0,9240

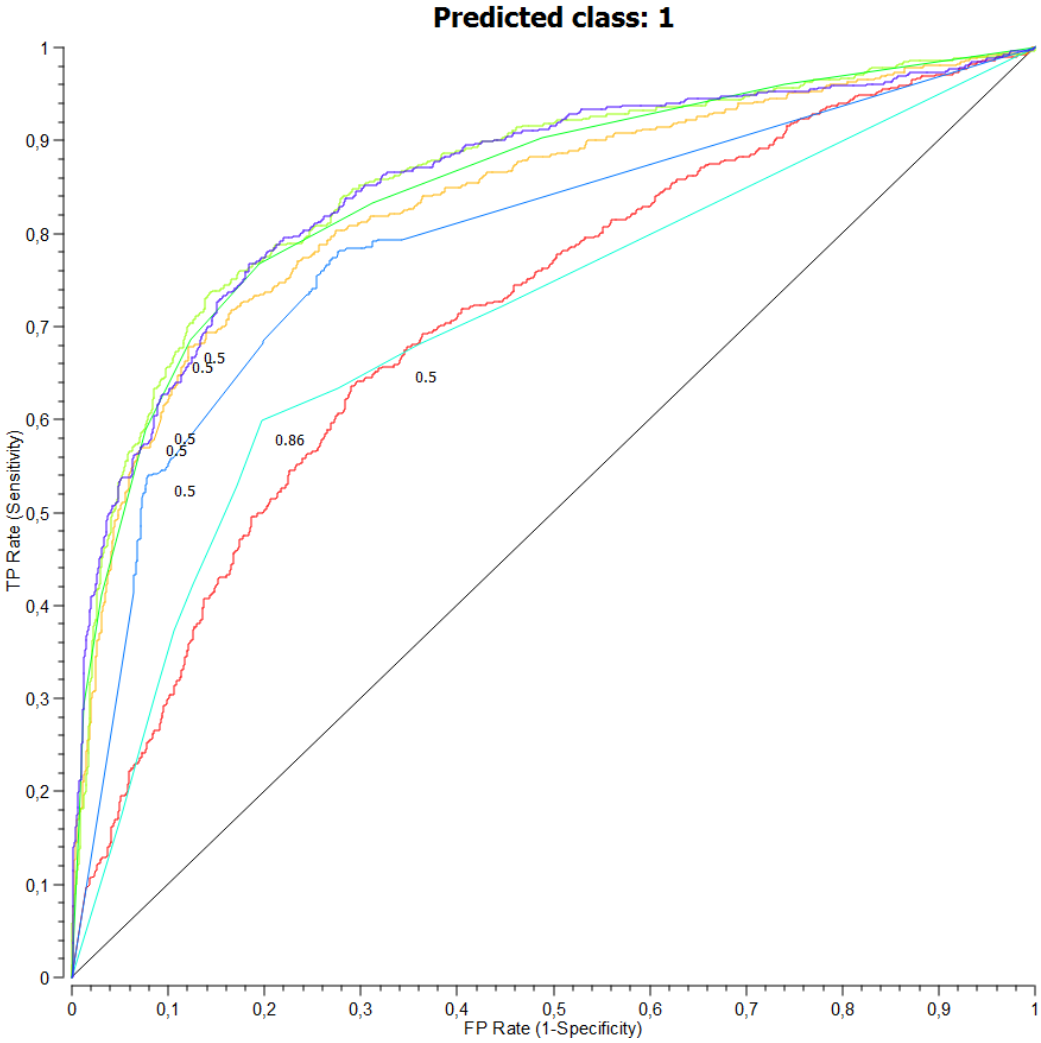
Tabela 4.5: Napovedna točnost, specifičnost in občutljivost napovednih modelov. R - ročno označene celice; CP1 - celice označene s programom CellProfiler; CP2 - celice označene s programom CellProfiler, brez mejnih celic.

Napovedni model	AUC	Natančnost	Specifičnost	Občutljivost
SVM	0,8426	0,7639	0,6081	0,8600
naključni gozdovi	0,8778	0,8186	0,7131	0,8838
naivni Bayes	0,7698	0,6969	0,6617	0,7186
nevronske mreže	0,8634	0,7908	0,6916	0,8520
CN2	0,8058	0,7835	0,6188	0,8851
boosting	0,5636	0,6000	0,8545	0,2727
bagging	0,6882	0,6000	0,8909	0,2045

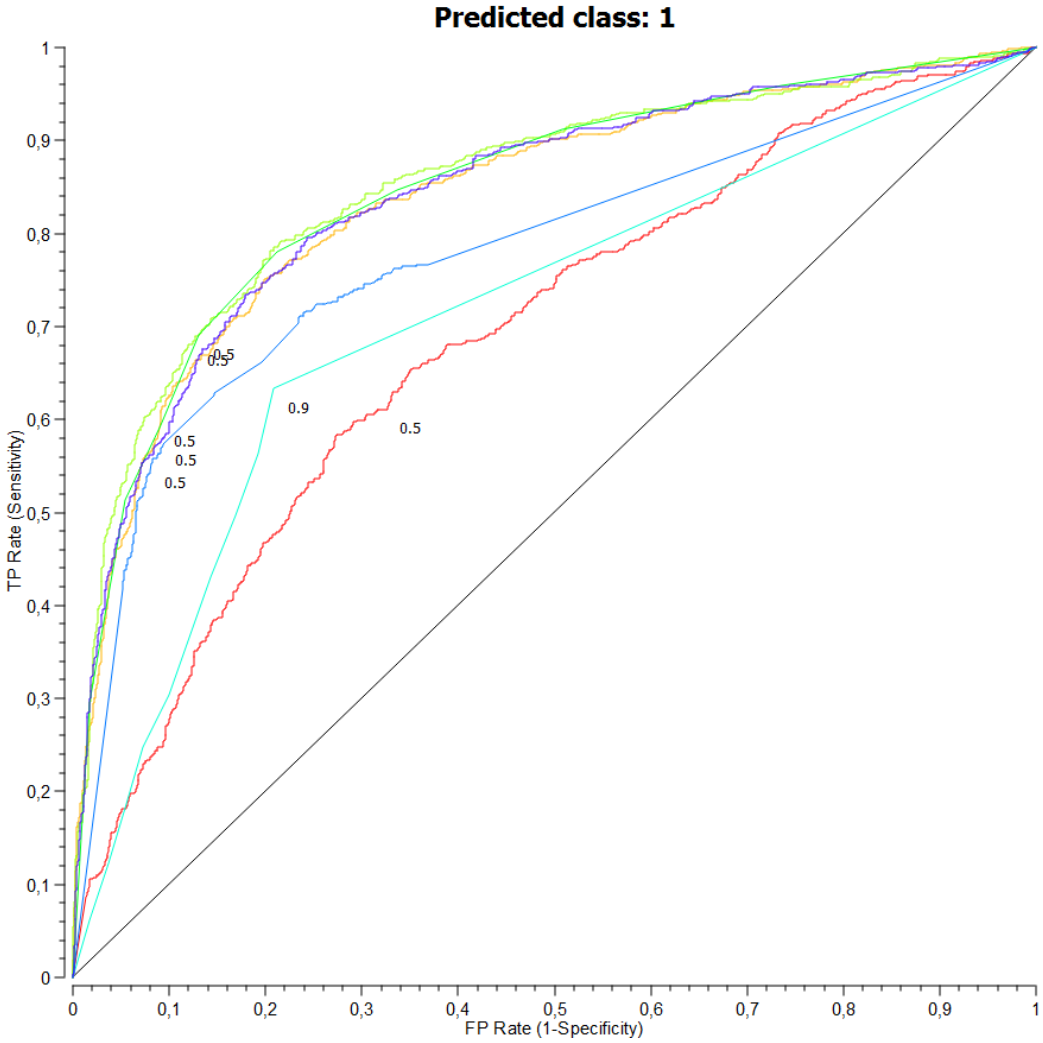
Tabela 4.6: Rezultati napovedovanja na ročno označenih slikah urinskih vzorcev (učenje na množici R).



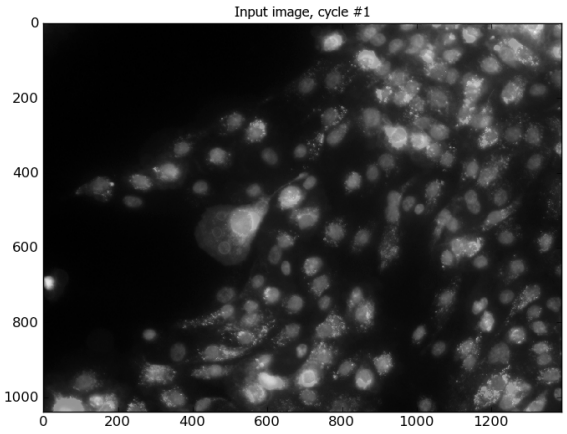
Slika 4.1: Krivulje ROC za množico ročno označenih celic (R). Barva krivulj: rdeča - naivni Bayes, rumena - SVM, svetlo zelena - nevronske mreže, zelena - bagging, turkizno modra - boosting, modra - CN2, vijolična - naključni gozdovi.



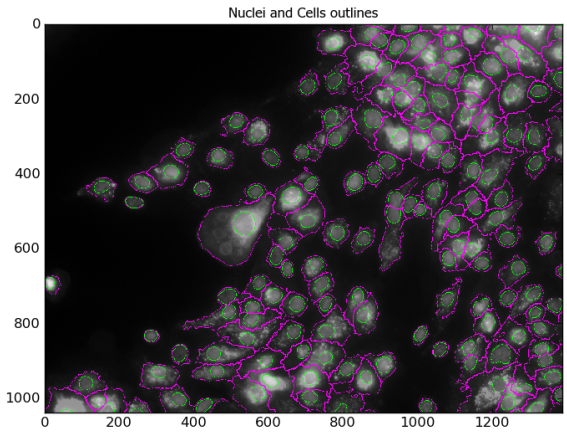
Slika 4.2: Krivulje ROC za množico celic označenih s programom CellProfiler (CP1). Barva krivulj: rdeča - naivni Bayes, rumena - SVM, svetlo zelena - nevronske mreže, zelena - bagging, turkizno modra - boosting, modra - CN2, vijolična - naključni gozdovi.



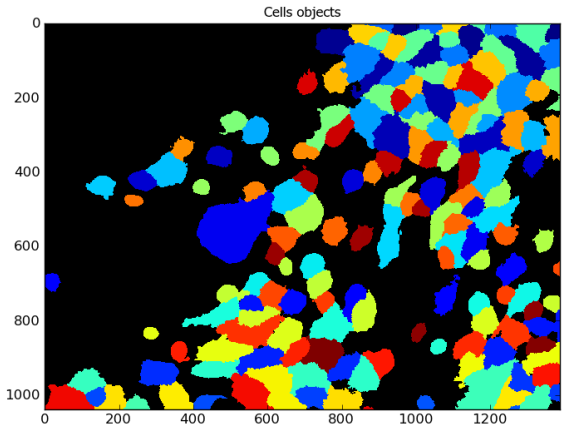
Slika 4.3: Krivulje ROC za množico celic označenih s programom CellProfiler, brez robnih celic (CP2). Barva krivulj: rdeča - naivni Bayes, rumena - SVM, svetlo zelena - nevronske mreže, zelena - bagging, turkizno modra - boosting, modra - CN2, vijolična - naključni gozdovi.



(a) Neoznačena slika.



(b) Označena jedra in membrane.



(c) Končni rezultat segmentacije.

Slika 4.4: Segmentacija celic z uporabo algoritma Watershed.

Poglavje 5

Sklepne ugotovitve in nadaljnje delo

Področje uporabe metod strojnega učenja in prepoznavanja slik v namene razumevanja in odkrivanja rakavih obolenj je zelo dejavno. Kljub temu veliko problemov ostaja neraziskanih, mnoge rešitve pa so raziskovalcem cenovno nedostopne. Eden izmed razlogov je tudi ta, da je vrst rakavih obolenj, kot tudi pristopov k reševanju problema, veliko. Problemski prostor je velik, generična rešitev pa tako rekoč ne obstaja. Uporaba statističnih in računalniških metod v navezi z domenskim znanjem medicine in biologije zahteva interdisciplinarne ekipe raziskovalcev iz različnih strok. Po drugi strani je dandanes širši množici dostopnih vse več orodij, ki ne zahtevajo poglobljenega tehničnega znanja.

Ključne ugotovitve:

- naš algoritem zelo dobro ločuje normalne in rakave urotelijske celice,
- algoritem je uspešnejši pri ročno označenih slikah,
- če pri samodejni segmentaciji odstranimo robne celice, to ne pripomore k boljšim rezultatom,

- dobra segmentacija celic je ključna za uspešnost nadaljnjih korakov algoritma,
- pri učenju in napovedovanju so najuspešnejše nevronske mreže, naključni gozdovi in bagging, ki dosegajo podobne rezultate,
- človek, ki označuje celice ročno, obliko celice velikokrat posploši,
- algoritem dosega slabše rezultate, če je celic več, so grupirane ali se celo prekrivajo,
- algoritem v tej obliki ni primeren za delo s citopatološkimi urinskimi vzorci.

Raziskovalci in zdravniki lahko s prilagajanjem parametrov posameznih korakov algoritem uporabijo v različne namene. Z umerjanjem občutljivosti in specifičnosti lahko v namen statistične obdelave dosežemo, da algoritem dovolj zanesljivo napove delež normalnih in rakavih urotelijskih celic. V primeru, da želimo strokovnjaka samo opozoriti na celice, kjer je verjetnost malignosti visoka, pa prilagodimo prag v prid občutljivosti in škodo specifičnosti. Opažamo, da je v postopku ključen korak obdelava mikroskopskih slik. Vsaka napaka pri zaznavi oblike in sestavnih delov celice v podatke vnese šum, kar neposredno vpliva na rezultate razločevanja. V sklopu razširitev in nadaljnjega dela zato priporočamo izboljšave algoritma, ki skrbi za samodejno segmentacijo. Med možnimi izboljšavami so natančnejše iskanje jeder, izločanje lažnih jeder in združevanje regij, kjer algoritem nepravilno razdeli celico na več delov. Izboljšave bi prav tako lahko vsebovale odkrivanje prekrivajočih se celic, tako da bi algoritem opazoval sliko tudi v globino. Razširitve omogoča tudi korak, ki v nabranih podatkih odkriva vzorce. Nabor obstoječih značilk bi lahko razširili z novimi ali pa kombinacijami obstoječih, preizkusili pa bi lahko še kakšen drug napovedni model, mogoče celo s področja nenadzorovanega učenja.

Omenili smo, da na tem področju obstaja veliko raziskav, ki se razlikujejo po podrobnostih pristopa, v opazovanem obolenju ali pa kombinaciji obeh. Naš pristop temelji na konceptih opisanih v podobnih raziskavah. Izsledki kažejo, da je naša implementacija uspešna pri reševanju problema samodejnega razločevanja normalnih in rakavih celic. Pristop s samodejno segmentacijo in samodejnim razločevanjem očitno skrajša čas pregledovanja in analize mikroskopskih slik. Naše mnenje je, da so pomemben prispevek tudi vse spremne statistične metrike in grafični prikazi, ki jih pri ročni obdelavi ni, ali pa so težje izračunljive. Doseženi rezultati so tako dobra motivacija za nadaljnji razvoj algoritma in pravočasnega odkrivanja in zdravljenja raka sečnega mehurja, kot tudi drugih rakavih obolenj.

Literatura

[1] D. A. Barocas, D. R. Globe, D. C. Colayco, A. Onyenwenyi, A. S. Bruno, T. J. Bramley, R. J. Spear, Surveillance and treatment of non-muscle-invasive bladder cancer in the USA, *Advances in Urology* 2012 (421709) (2012) 802 – 811.

[2] D. J. DeGraff, J. M. Cates, J. R. Mauney, P. E. Clark, R. J. Matusik, R. M. Adam, When urothelial differentiation pathways go wrong: Implications for bladder cancer development and progression, *Urologic Oncology: Seminars and Original Investigations* 31 (6) (2013) 802 – 811. doi:http://dx.doi.org/10.1016/j.urolonc.2011.07.017.
URL http://www.sciencedirect.com/science/article/pii/S1078143911002377

[3] P. Korošec, W. de Mello Jr, K. Jezernik, et al., Differentiation of epithelial cells in the urinary tract, Cell and tissue research 320 (2) (2005) 259–268.

[4] M. E. Kreft, S. Hudoklin, K. Jezernik, R. Romih, Formation and maintenance of blood–urine barrier in urothelium, *Protoplasma* 246 (1) (2010) 3–14. doi:10.1007/s00709-010-0112-1.

[5] E. Lasič, T. Višnjar, M. E. Kreft, *Reviews of Physiology, Biochemistry and Pharmacology*, Springer International Publishing, Cham, 2015, Ch. Properties of the Urothelium that Establish the Blood–Urine Barrier and Their Implications for Drug Delivery, pp. 1–29. doi:10.1007/112_

2015_22.
URL http://dx.doi.org/10.1007/112_2015_22

[6] Spletišče za dostop do podatkov o raku v sloveniji in drugod, dostopano: 15.9.2016.
URL http://www.slora.si/analizaslo

[7] J. A. Cruz, D. S. Wishart, Applications of machine learning in cancer prediction and prognosis.

[8] S. Chen, M. Zhao, G. Wu, J. Yao, Chunyan nad Zhang, Recent advances in morphological cell image analysis, *Computational and Mathematical Methods in Medicine* 2012. doi:10.1155/2012/101536.

[9] J. B. Roerdink, A. Meijster, The watershed transform: Definitions, algorithms and parallelization strategies, *Fundamenta Informaticae* 41 (2001) 187 – 228.

[10] S. Beucher, C. Lantuejoul, Use of watersheds in contour detection, International Workshop on image processing: Real-time Edge and Motion detection/estimation (17.-21., 1979).

[11] A. E. e. a. Carpenter, Cellprofiler: image analysis software for identifying and quantifying cell phenotypes, *Genome Biol* 7 (10). doi:10.1186/gb-2006-7-10-r100.

[12] C. O. D. Solórzano, S. Costes, D. Callahan, B. Parvin, M. Barcellos-Hoff, Applications of quantitative digital image analysis to breast cancer research, *Microscopy Research and Technique* 59 (2002) 119 – 127.

[13] D. Glotsos, P. Spyridonos, D. Cavouras, P. Ravazoula, P. A. Dadioti, G. Nikiforidis, An image-analysis system based on support vector machines for automatic grade diagnosis of brain-tumour astrocytomas in clinical routine, *Medical Informatics and the Internet in Medicine* 30 (2005) 179 – 193. doi:10.1080/14639230500077444.

[14] T. Tikkanen, P. Ruusuvuori, L. Latonen, H. Huttunen, Training based cell detection from bright-field microscope images, in: *Image and Signal Processing and Analysis (ISPA), 2015 9th International Symposium on*, IEEE, 2015, pp. 160–164.

[15] M. Wang, X. Zhou, F. Li, J. Huckins, R. W. King, S. T. Wong, Novel cell segmentation and online svm for cell cycle phase identification in automated microscopy, *Bioinformatics* 24 (1) (2008) 94–101.

[16] M. Hrebień, P. Steć, T. Nieczkowski, A. Obuchowicz, Segmentation of breast cancer fine needle biopsy cytological images, *International Journal of Applied Mathematics and Computer Science* 18 (2) (2008) 159–170.

[17] L. J. Belaid, W. Mourou, Image segmentation: a watershed transformation algorithm, *Image Anal Stereol* 28 (2009) 93 – 102.

[18] R. Warren, R. E. Smith, A. K. Cybenko, Use of Mahalanobis distance for detecting outliers and outlier clusters in markedly non-formal data: a vehiclar traffic example.

[19] E. Joakim, Mahalanobis' distance beyond normal distributions, dostopano: 16.2.2016.
URL `https://www.researchgate.net/publication/265259428_MAHALANOBIS'_DISTANCE_BEYOND_NORMAL_DISTRIBUTIONS`

[20] N. Otsu, A threshold selection method from gray-level histograms, *Automatica* 11 (285-296) (1975) 23–27.

[21] I. H. Witten, E. Frank, M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd Edition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.

[22] M. Lenič, Multimetodna gradnja klasifikacijskih sistemov, doktorska disertacija; Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru (november 2003).

[23] L. Breiman, Random forests, *Machine Learning* 45 (1) (2001) 5–32. doi:10.1023/A:1010933404324.

[24] J. D. et al., Orange: Data mining toolbox in python, *Journal of Machine Learning Research* 14 (2013) 2349–2353.
URL http://jmlr.org/papers/v14/demsar13a.html

[25] Cell staining in microscopy: Types, techniques, preparations and procedures, dostopano: 1.3.2016.
URL http://www.microscopemaster.com/cell-staining-microscopy.html

[26] Dapi (4′,6-diamidino-2-phenylindole), dostopano: 1.3.2016.
URL https://www.thermofisher.com/si/en/home/life-science/cell-analysis/fluorophores/dapi-stain.html?icid=fr-dapi-main

[27] Dil stain (1,1'-dioctadecyl-3,3,3',3'-tetramethylindocarbocyanine perchlorate ('dii'; diic18(3))), dostopano: 1.3.2016.
URL https://www.thermofisher.com/order/catalog/product/D282

[28] Dio'; dioc18(3) (3,3'-dioctadecyloxacarbocyanine perchlorate), dostopano: 1.3.2016.
URL https://www.thermofisher.com/order/catalog/product/D275

[29] F. Fleuret, T. Lic, C. Dubout, K. E. Wampler, S. Yantis, D. Geman, Comparing machines and humans on a visual categorization test, *PNAS* 108 (2011) 17621–17625.

[30] A. Borji, L. Itti, Human vs. computer in scene and object recognition, in: *Computer Vision and Pattern Recognition (CVPR)*, 2014 IEEE Conference on, 2014, pp. 113–120. doi:10.1109/CVPR.2014.22.

[31] M. Vorobyov, Shape classification using zernike moments, Tech. rep., Technical Report. iCamp-University of California Irvine (2011).

[32] H. Hse, A. R. Newton, Sketched symbol recognition using zernike moments, in: *Pattern Recognition. ICPR 2004. Proceedings of the 17th International Conference on*, Vol. 1, IEEE, pp. 367–370.

[33] S. Sisakhtnezhad, L. Khosravi, Emerging physiological and pathological implications of tunneling nanotubes formation between cells, *European journal of cell biology* 94 (10) (2015) 429–443.